Training – Day #9

15 June 2023 06:42

#Dashboard

Dashboard = collection of panels

2 Type of Dashboard builder :

(1) **Classic Type** :
- (i) generated on top of **XML**
- (ii) from starting
- (iii) can integrate with HTML, CSS, JS

(A) **Static Classic Dashboard**
- (i) No option of input from user.

(B) **Dynamic Classic Dashboard**
- (i) Option of input from user. User can enter/filter specific value

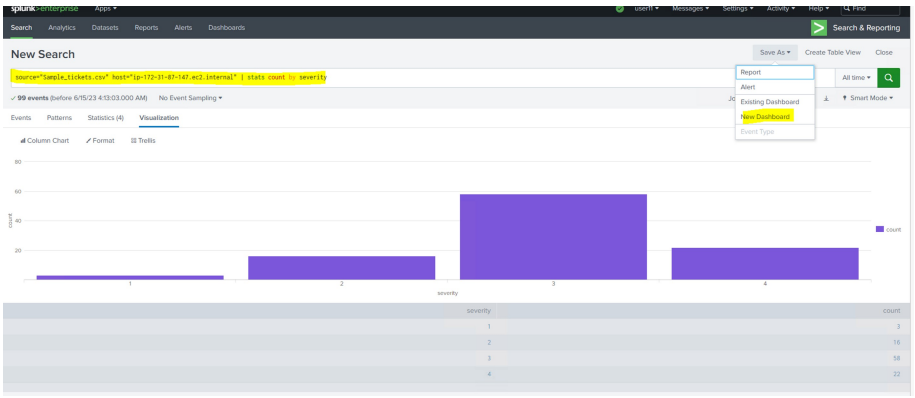
(2) ***Studio Type :***
- (i) generated on top of **JASON**
- **(ii)** from v8.2.x
- (iii) ITSI **Glass Table** Generation

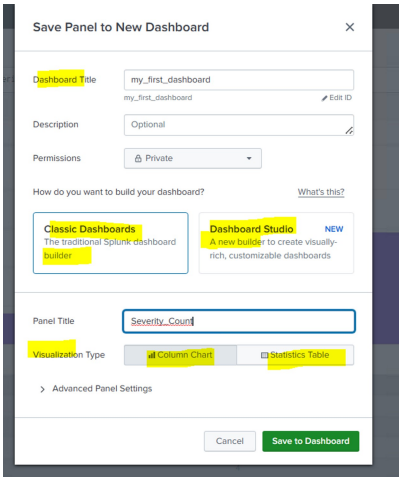
3 Optimization Process for Dashboard (in both types):
- (1) Base Search
- (2) Saved Search
- (3) Summary Index

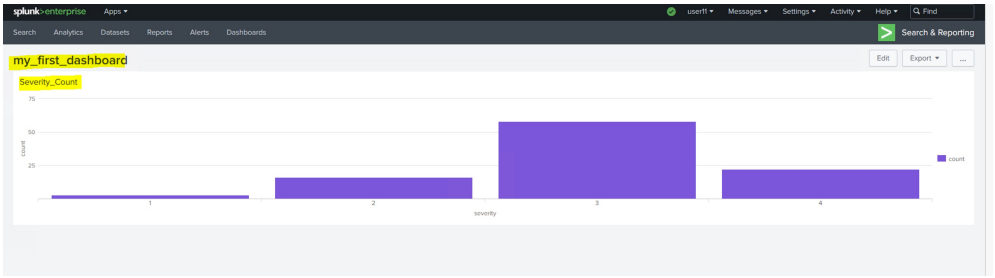
?

How to create a new Dashboard ?

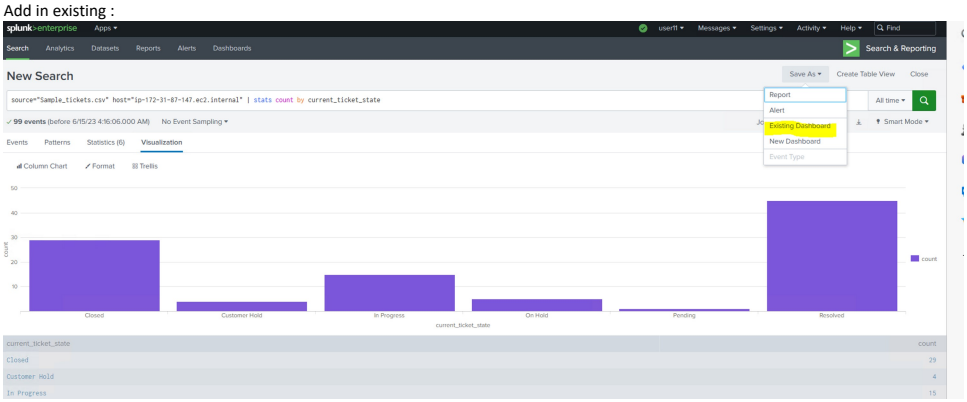
Write a Search query -> save as **New** Dashboard (we can add in an **Existing** Dashboard also)

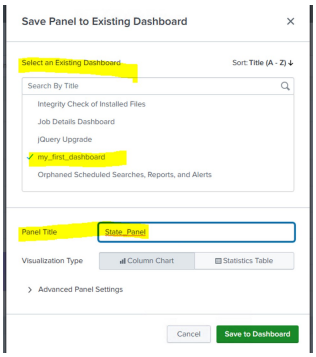


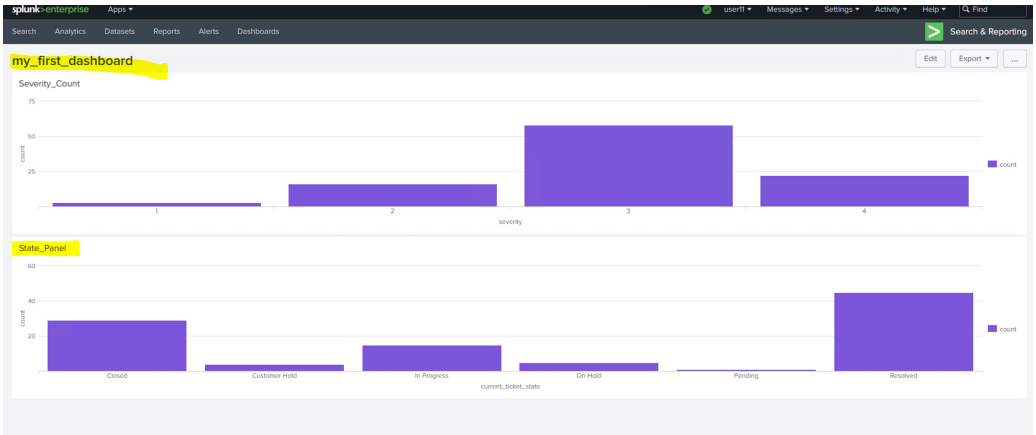




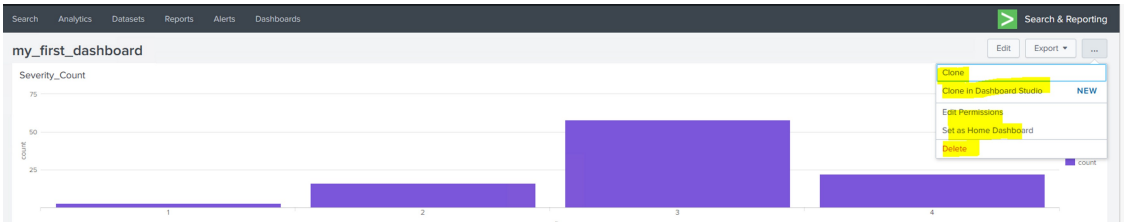
Add in existing :

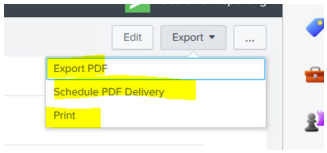




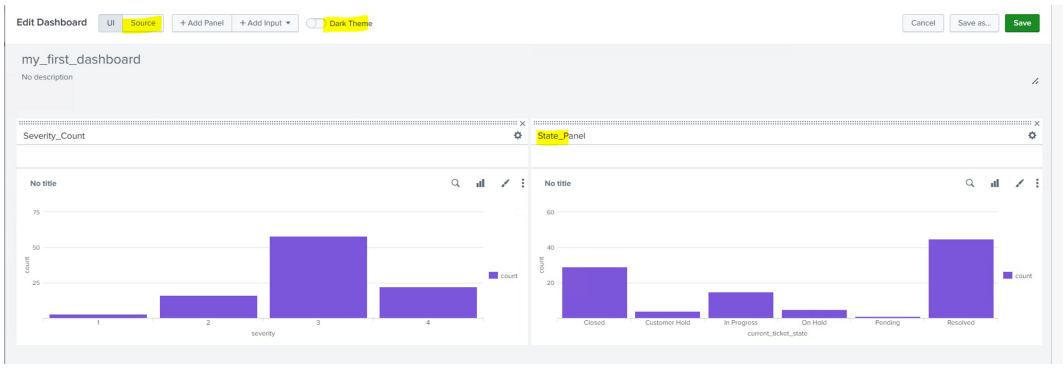


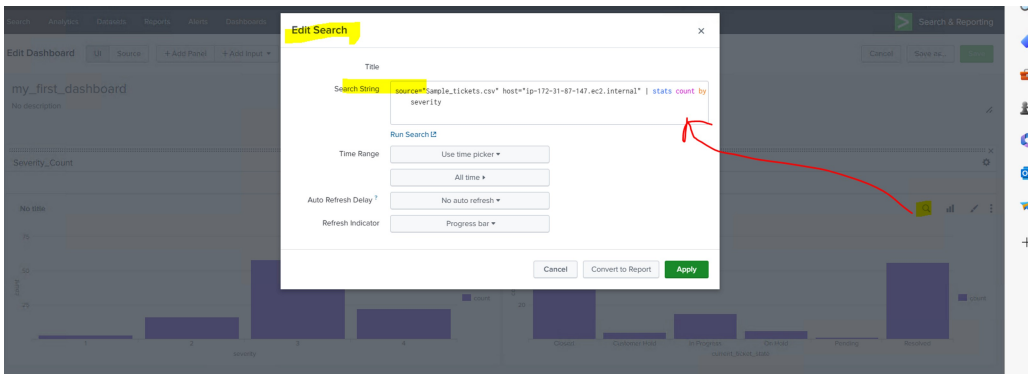
Options:

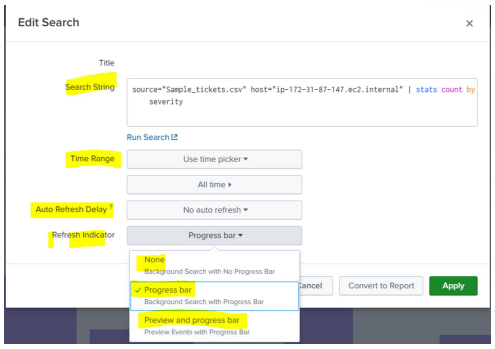


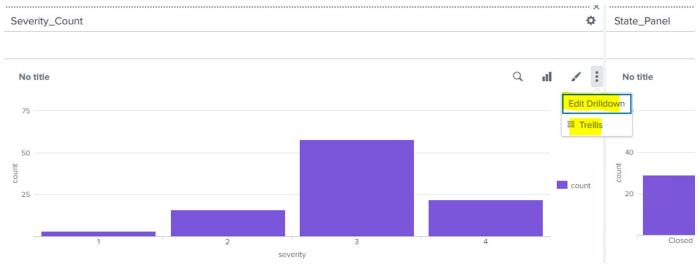


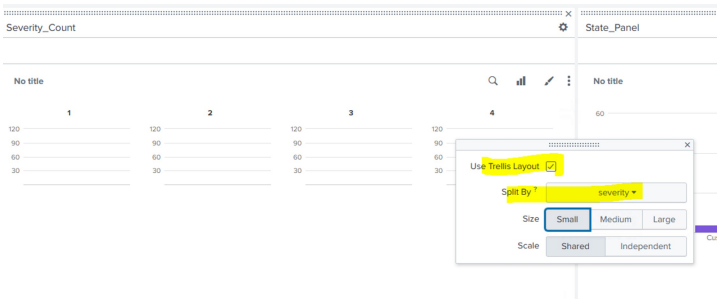
Edit => Drag the Panels to adjust:



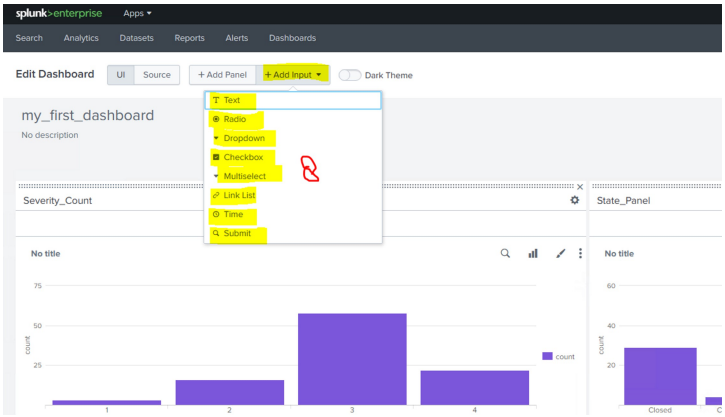


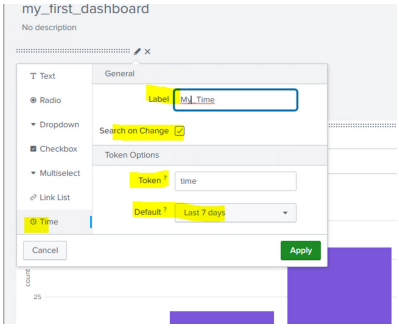




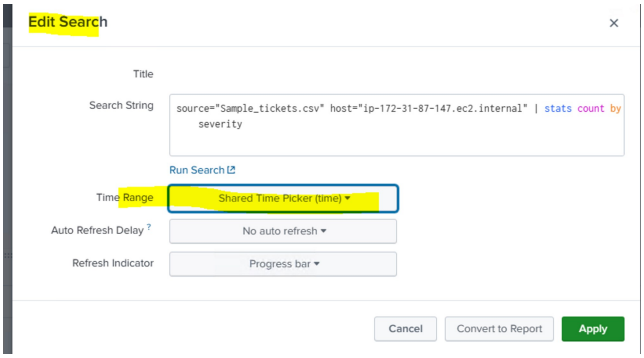


Add-Inputs => 8 Types of Inputs

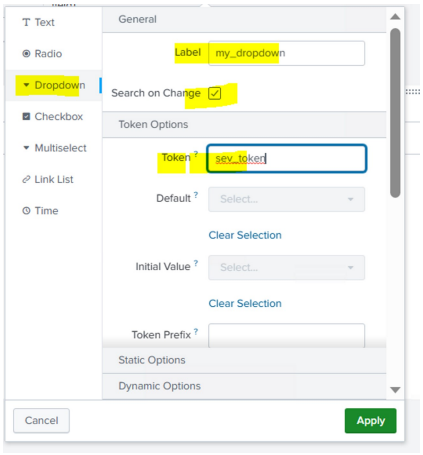


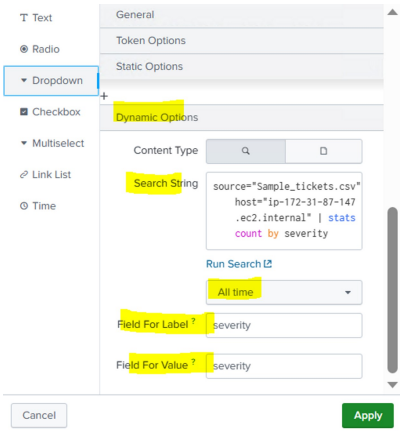


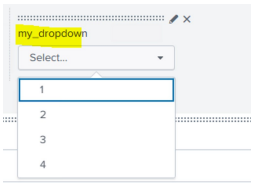
Now go to each Panel => and select **"Shared Time Picker (time)"**

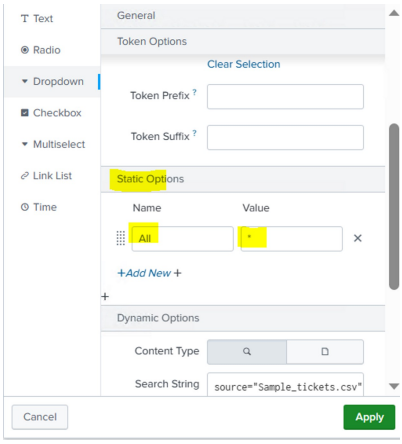


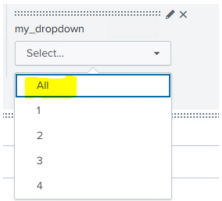
Creating Dropdown input :



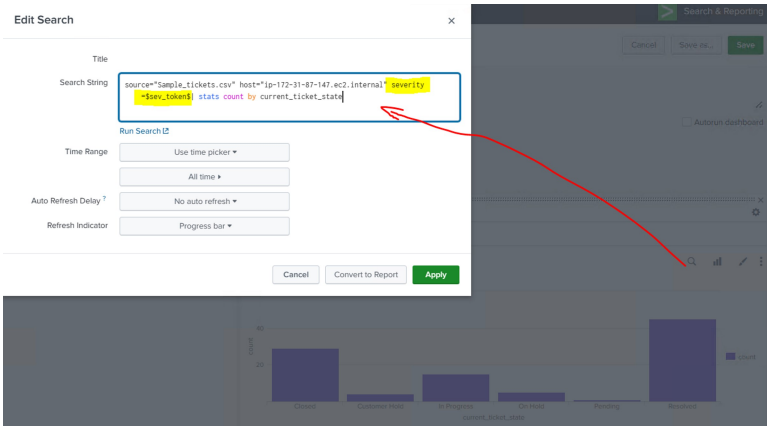


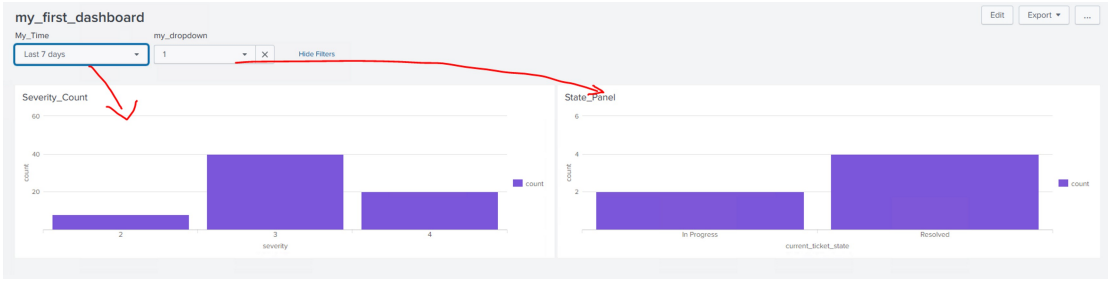




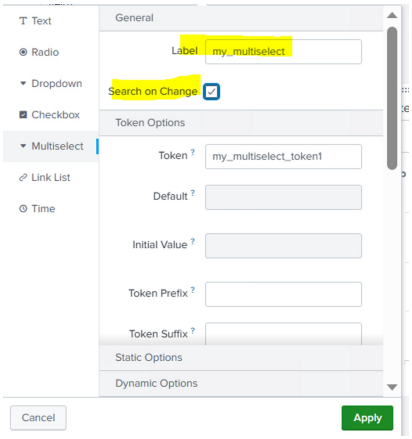


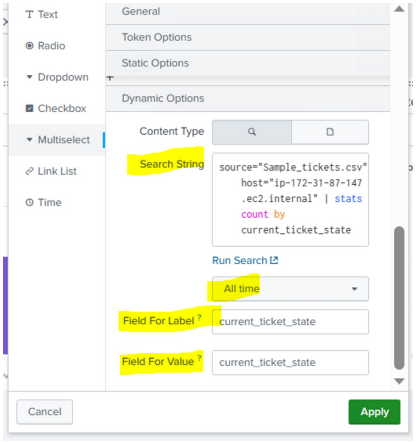
Mapping dropdown token to dashboard panel :

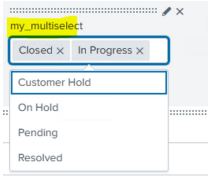




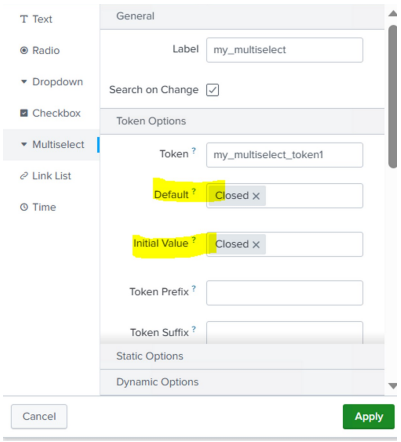
Creating Multiselect Input :



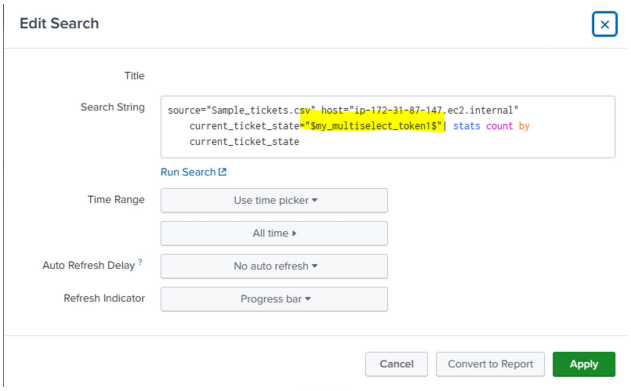




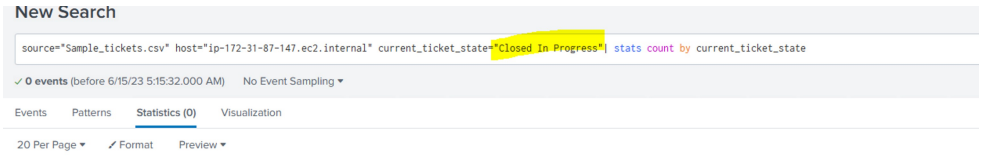
Again edit and select Default/Initial Value



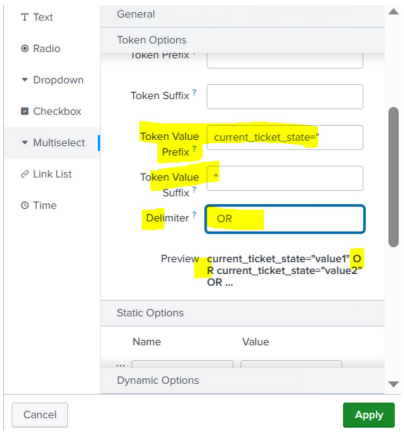
Map with Panel :



No result as values are not separate



To fix this => Use token_prefix, token suffix, and delimiter :



Note : check the Space in Preview and adjust Space before after "Delimiter"

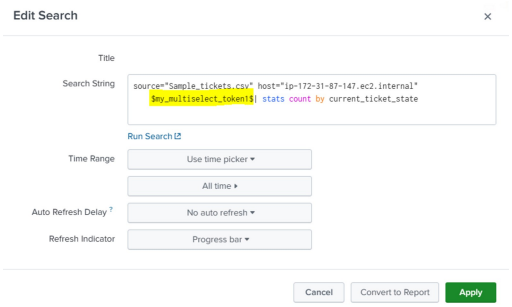
Modify the search query on panel :

Before :

source="Sample_tickets.csv" host="ip-172-31-87-147.ec2.internal"
current_ticket_state="$my_multiselect_token1$"| stats count by current_ticket_state

After Modification :

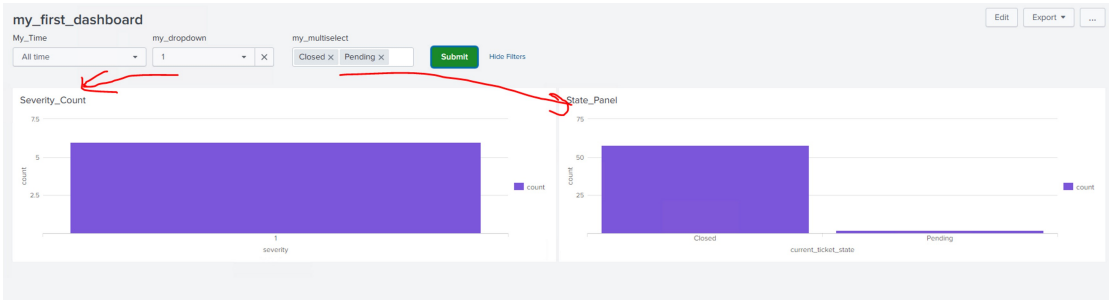
source="Sample_tickets.csv" host="ip-172-31-87-147.ec2.internal" "$my_multiselect_token1$|
stats count by current_ticket_state



"SUBMIT" button :

It is used to pass values to panel searches all at a time instead of every change.

Unset all the input "Search on change" and Add a **Submit** button in last:



#Dasboard Optimization :

(1) ***Base Search*** :

Base search **create a Virtual table** and from this virtual table we can extract/pull the data in the respective panels/dropdown.

Step #1 : Go to "Source" and add below XML to create base search

```
<search id ="base_search1">
  <query>
       index=main Sample_tickets.csv" host="ip-172-31-87-147.ec2.internal" | table severity,
      current_ticket_state
</query>
  <earliest>0</earliest>
  <latest>now</latest>
 </search>
```

Step #2 : Edit the <search> and <query> in panel

Edit Dashboard | UI | Source

8 Validation warnings

```
<form version="1.1">
  <label>my_first_dashboard</label>

  <search id ="base_search1">

    <query>

      index=main Sample_tickets.csv" host="ip-172-31-87-147.ec2.internal" | table severity, current_ticket_state

    </query>

    <earliest>0</earliest>
    <latest>now</latest>
  </search>


  <fieldset submitButton="true" autoRun="false">
    <input type="time" token="time" searchWhenChanged="false">
      <label>My_Time</label>
      <default>
        <earliest>-7d@h</earliest>
        <latest>now</latest>
      </default>
    </input>

    <input type="dropdown" token="sev_token" searchWhenChanged="false">
      <label>my_dropdown</label>
      <choice value="*">All</choice>
      <fieldForLabel>severity</fieldForLabel>
      <fieldForValue>severity</fieldForValue>
      <search base="base_search1">
        <query>| stats count by severity</query>
        <earliest>0</earliest>
        <latest></latest>
      </search>
    </input>

    <input type="multiselect" token="my_multiselect_token1" searchWhenChanged="false">
      <label>my_multiselect</label>
      <default>Closed</default>
      <initialValue>Closed</initialValue>
      <fieldForLabel>current_ticket_state</fieldForLabel>
      <fieldForValue>current_ticket_state</fieldForValue>
      <search base="base_search1">
```

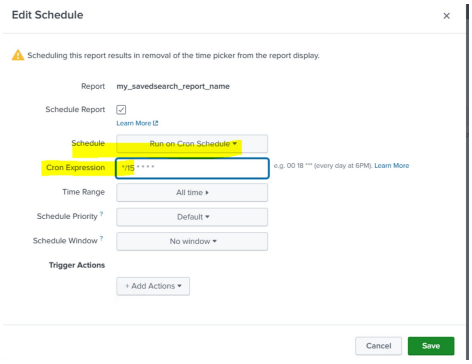
NOTE :
(1) base search query should be transformational
(2) base search will generate virtual table at each refresh of dashboard.

(1) ***Saved Search*** :

Saved search will also **create a Virtual table, but it will not be created every refresh of dashboard but we have to define a time-interval** like evry 15 min or 30 min and virtual table will be created at that time-interval.

After 7.x Splunk has merged the Saved Search with Report generation

Step #1 : Create a Report let's name "**my_savedsearch_report_name"** and schedule its time interval.



Step #2 : In Dashboard add in xml

<search id="saved_search" ref="my_savedsearch_report_name"></search>

```
<form version="1.1">
  <label>my_first_dashboard</label>

  <search id ="base_search1">

    <query>

      index=main Sample_tickets.csv" host="ip-172-31-87-147.ec2.internal" | table severity, current_ticket_state

    </query>

    <earliest>0</earliest>
    <latest>now</latest>
  </search>

<search id="saved_search" ref="my_savedsearch_report_name"></search>

  <fieldset submitButton="true" autoRun="false">
    <input type="time" token="time" searchWhenChanged="false">
      <label>My_Time</label>
      <default>
        <earliest>-7d@h</earliest>
        <latest>now</latest>
      </default>
    </input>

    <input type="dropdown" token="sev_token" searchWhenChanged="false">
      <label>my_dropdown</label>
      <choice value="*">All</choice>
```

(3) Summary Index :

It will writes the output of a search query into a **separate index.**

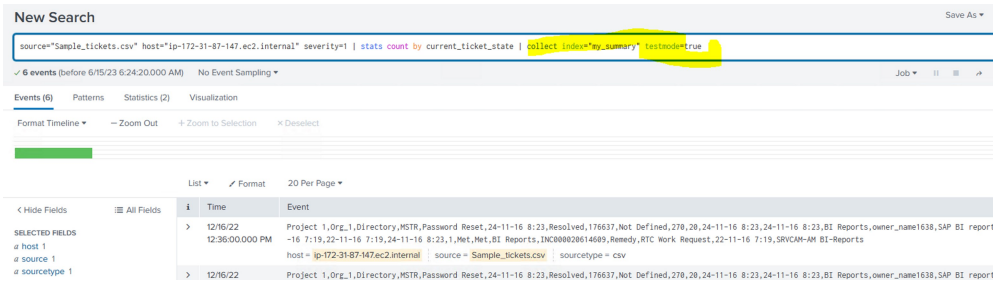
Downside : more resource consumption but **will not cost license as sourcetype as** "**stash**"

We can manually push the data
Or we can use automatic one.

(1) Manual creation of Summary-Index:

Step #1 : create an index
Step #2 : prepare search query and add " ***| collect index="<new index name> "testmode=true***"

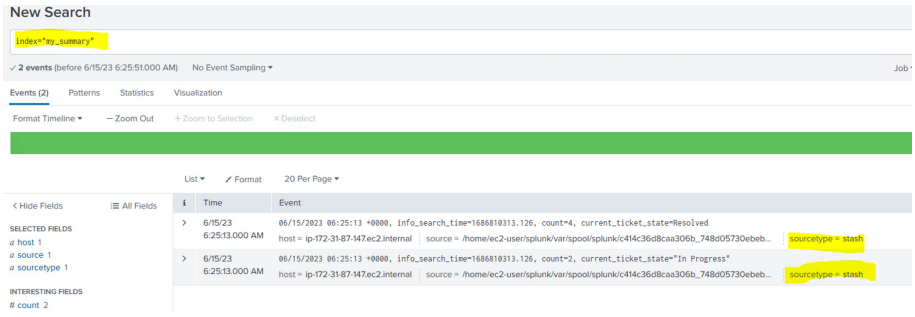
testmode= true => it will only show the data but will not push the data into index
testmode = false => it will push the data into index



Step #3 : prepare search query and add " ***| collect index="<new index name>" testmode=false***"

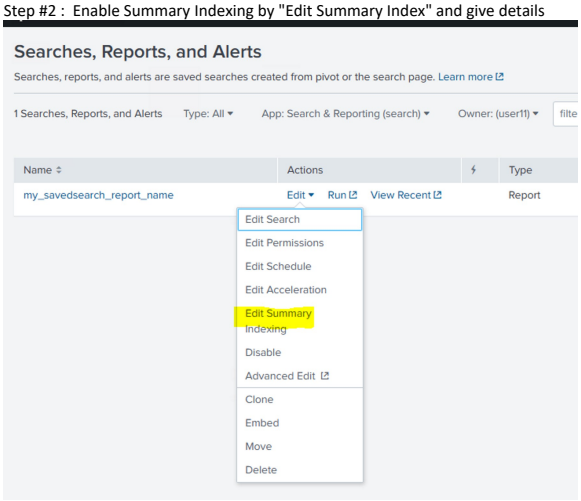
```
source="Sample_tickets.csv" host="ip-172-31-87-147.ec2.internal" severity=1 | stats count by current_ticket_state | collect index="my_summary" testmode=false
```

Data will get saved into index with sourcetype as "stash"



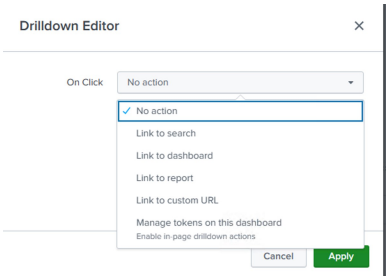
(2) Automated creation of Summary-Index:

Step #1 : Create a Report Settings=> Report => Edit Reprot
Step #2 : Enable Summary Indexing by "Edit Summary Index" and give details



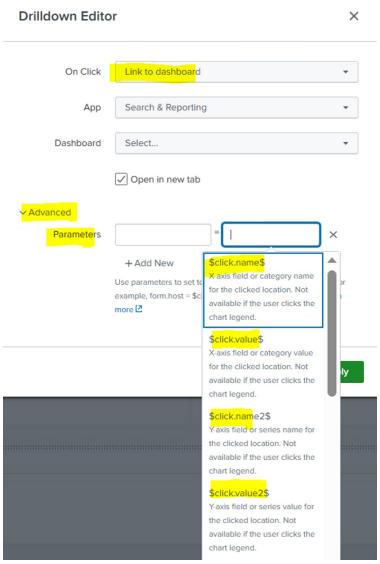
---
(1) Drilldown in Dashboard :

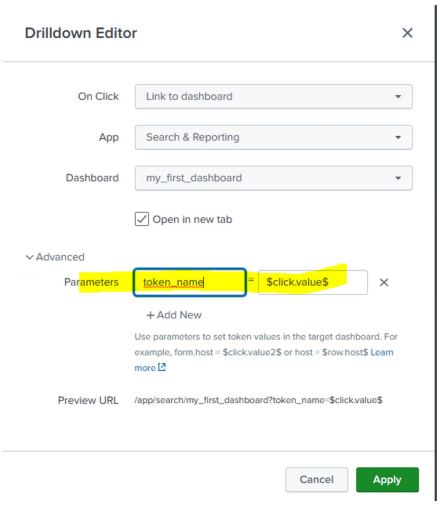
If I will click any graph/column then it will show the backend events in separate page.



❓
Question : How to pass the value from One dashboard to other ?
Answer : Use drilldown => "Link to dashboard"

Click on Name
Click on Value



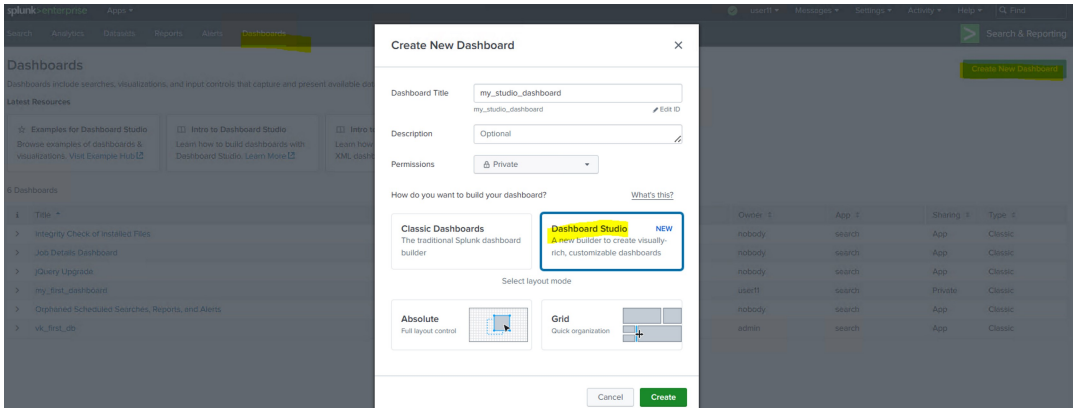


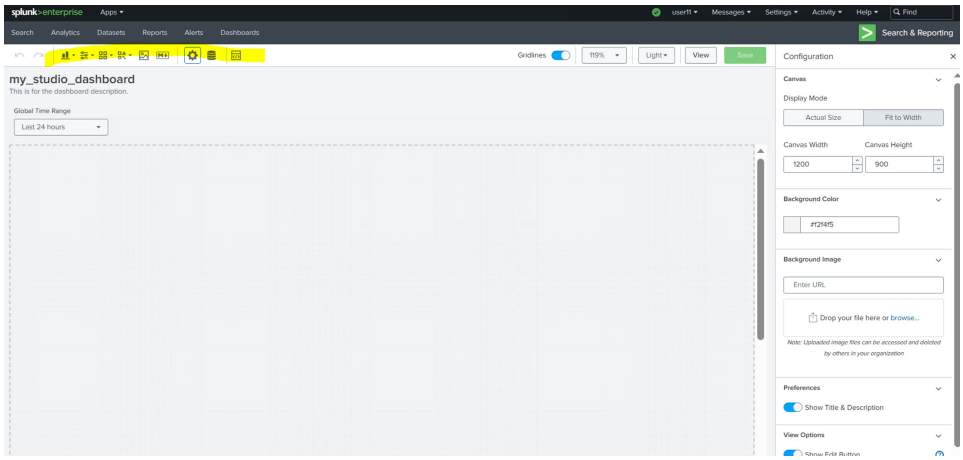
---

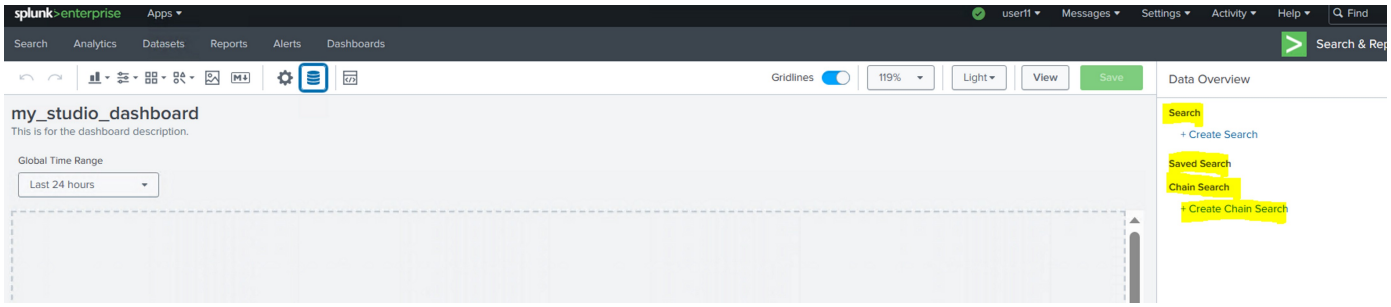
(2) STUDIO Dashboard:

1. Backend query is in JSON
2. Can add Background Image (**those images are get stored in KV store**)
3. **integration not possible** with HTML, CSS

❓
How to create a Studio Dashboard ?

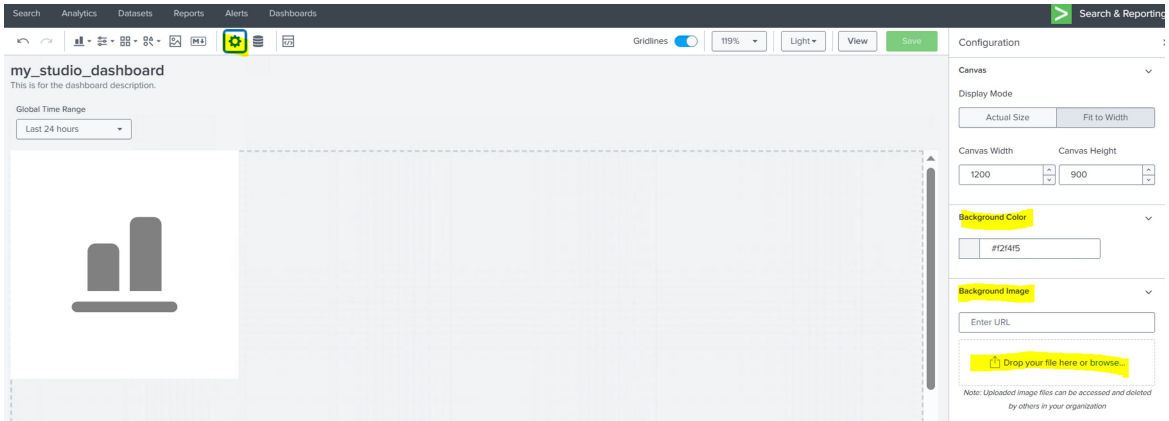


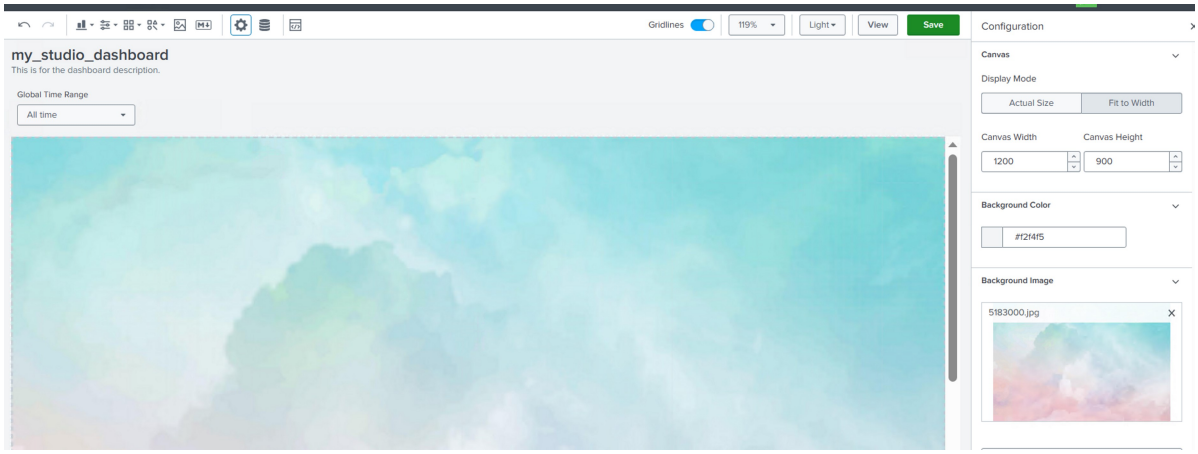




Difference between Search and Chain Search ??

Settings Background via Canvas :

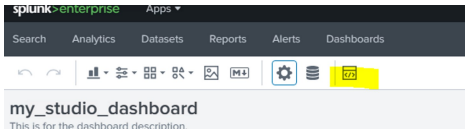


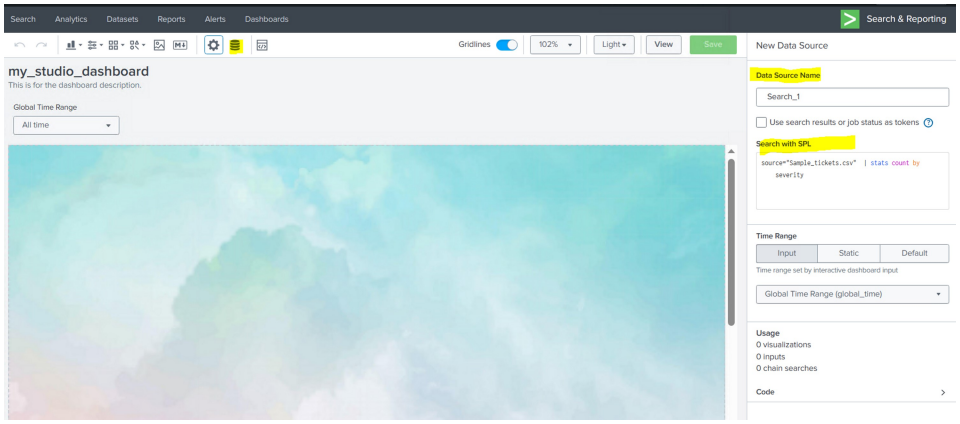


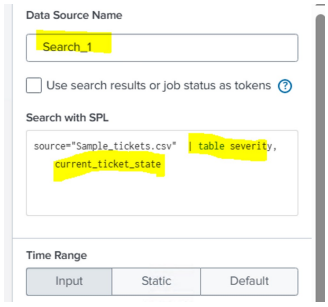
Background image is getting stored in kvstore:

```
    },
    "layout": {
        "type": "absolute",
        "options": {
            "display": "auto-scale",
            "backgroundImage": {
                "sizeType": "contain",
                "x": 0,
                "y": 0,
                "src": "splunk-enterprise-kvstore://648ab5eecb5f9b051f2a7a65"
            }
        },
        "structure": [],
        "globalInputs": [
            "input_global_trp"
        ]
```

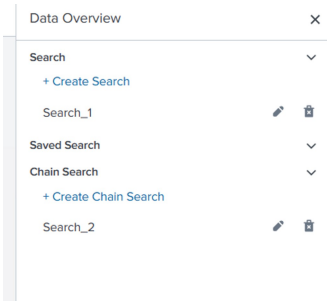
Check the JSON :



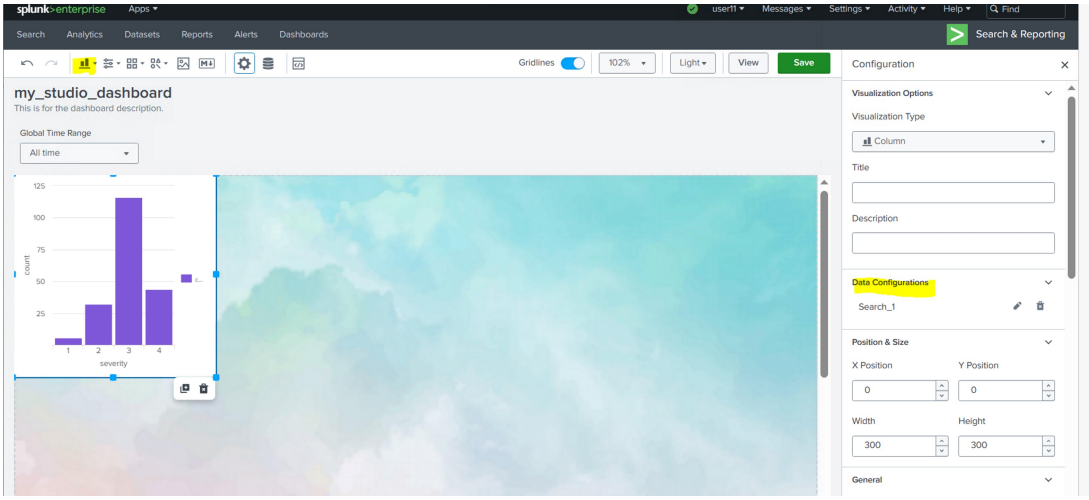




Search should be like base search => table command



Select a Graph => click Cancel => then select "Data Configuration"



JSON:

Note : There is no save button. If any modification is done, it will not allow us to go back until we remove that modification from JSON

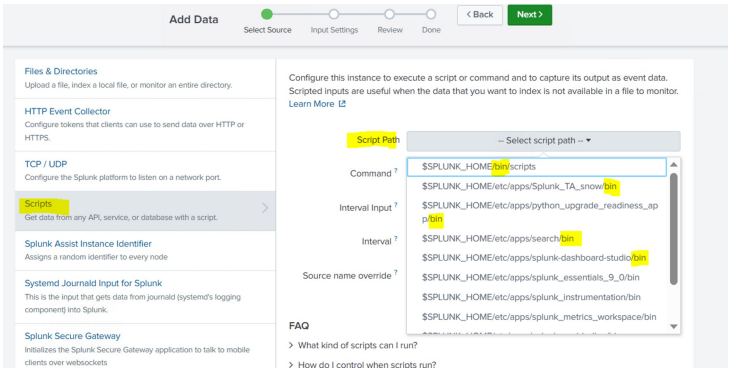
#Scripted Input

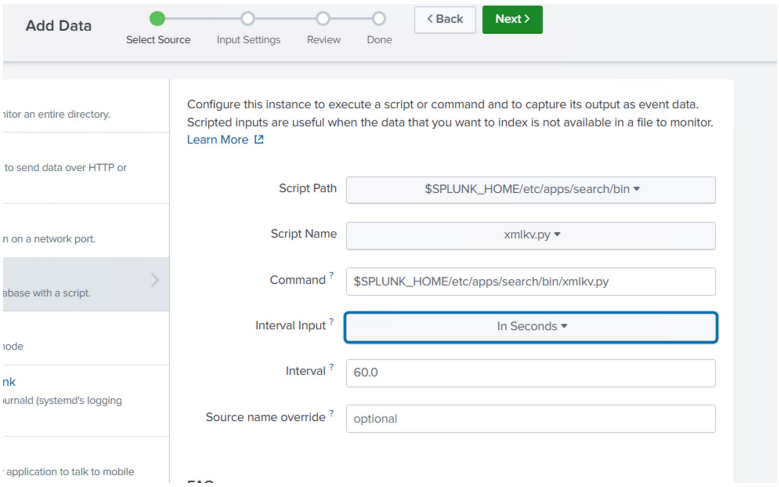
Data ingestion will done through an script.

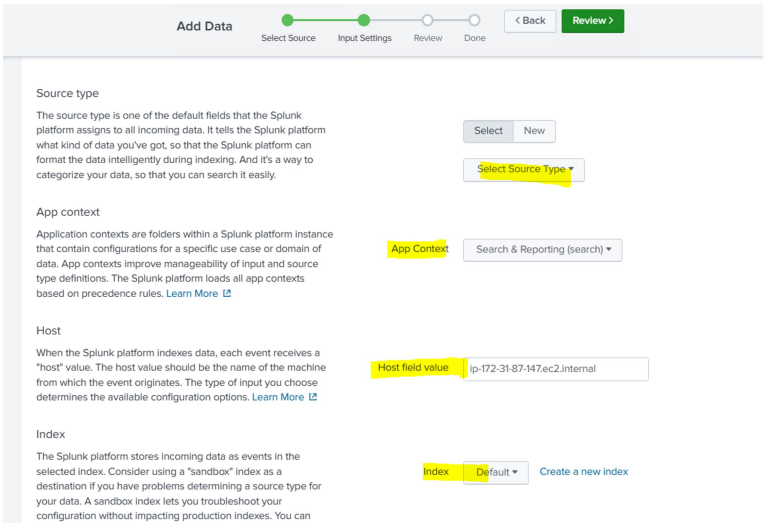
TMDB website for movie details => create account and get API key

[The Movie Database (TMDB) (themoviedb.org)]

Note :
(1) script can be anywhere but it must be in **/bin** folder (if SAAS, then store the script in Heavy Forwarder)







#Index Time Field Extraction

Pull the event at the time of insertion into index