Splunk Classic Dashboard – Detailed Notes

1) What is a Classic Dashboard (Simple XML)?

Classic Dashboards are the legacy, XML-based dashboards in Splunk Web (pre-Dashboard Studio). They're panel-driven, easy to build, support tokens, drilldowns, and base/post-process searches, and are perfect for quick operational views.

When to use: quick KPI boards, forms with inputs/tokens, drilldowns into searches.
When to prefer Studio: rich visuals, custom layout/styling, modern theming.

2) Anatomy of a Classic Dashboard

```
<dashboard>
  <label>My Classic Dashboard</label>
  <description>High-level system overview</description>

  <!-- (Optional) Form inputs create tokens -->
  <form>
    <input type="time" token="tspan">
      <label>Time Range</label>
      <default>
        <earliest>-24h@h</earliest><latest>now</latest>
      </default>
    </input>
    <input type="dropdown" token="sourcetype_tok">
      <label>Sourcetype</label>
      <choice value="*">All</choice>
      <choice value="access_combined">access_combined</choice>
      <default>*</default>
    </input>
  </form>

  <row>
    <panel>
      <chart>
        <title>Events Over Time</title>
        <search>
          <query>index=main sourcetype="$sourcetype_tok$"
                 | timechart count</query>
          <earliest>$tspan.earliest$</earliest>
          <latest>$tspan.latest$</latest>
        </search>
        <option name="charting.chart">line</option>
```

```
        <option name="refresh.display">progressbar</option>
        <option name="refresh.auto.interval">60</option>
      </chart>
    </panel>

    <panel>
      <table>
        <title>Top Hosts</title>
        <search base="b1">
          <query>| top host limit=10</query> <!-- post-process -->
        </search>
      </table>
    </panel>
  </row>

  <!-- Base search feeding multiple post-process panels -->
  <search id="b1">
    <query>index=main sourcetype="$sourcetype_tok$"</query>
    <earliest>$tspan.earliest$</earliest>
    <latest>$tspan.latest$</latest>
  </search>
</dashboard>
```

3) Creating a Classic Dashboard (Step-by-Step)

1. Apps → Search & Reporting → Dashboards → Create New Dashboard → Classic (XML)

2. Add panels: from a search (Save As → Dashboard Panel) or Edit the dashboard and add panels in UI/XML.

3. Add form inputs (dropdown, text, time) to parameterize searches.

4. Use base + post-process for performance.

5. Set permissions (Private/App/Global) and schedule backing reports if needed.

4) Common Panel Examples (copy-paste ready)

A) Status code trend (last 24h)

```
<panel>
  <chart>
    <title>Status Codes Trend (24h)</title>
    <search>
      <query>index=web sourcetype=access_combined
             | timechart count by status</query>
      <earliest>-24h@h</earliest><latest>now</latest>
    </search>
    <option name="charting.chart">column</option>
```

```
  </chart>
</panel>
```

B) KPI single-value with sparkline

```
<panel>
  <single>
    <title>Total Errors</title>
    <search>
      <query>index=web status>=400
             | timechart count
             | eventstats latest(count) as now
             | eval spark=count</query>
      <earliest>-24h@h</earliest><latest>now</latest>
    </search>
    <option name="underLabel">last 24h</option>
    <option name="sparkline">true</option>
  </single>
</panel>
```

C) Base + two post-process tables

```
<search id="b_errors">
  <query>index=web status>=400</query>
  <earliest>$tspan.earliest$</earliest><latest>$tspan.latest$</latest>
</search>

<row>
  <panel>
    <table>
      <title>Top 10 URLs with Errors</title>
      <search base="b_errors">
        <query>| top limit=10 uri_path</query>
      </search>
    </table>
  </panel>
  <panel>
    <table>
      <title>Top 10 Clients with Errors</title>
      <search base="b_errors">
        <query>| top limit=10 clientip</query>
      </search>
    </table>
  </panel>
</row>
```

D) Drilldown to another dashboard/search

```
<panel>
  <table>
```

```
    <title>Top Clients (Drilldown)</title>
    <search>
      <query>index=web | top clientip</query>

<earliest>$tspan.earliest$</earliest><latest>$tspan.latest$</latest>
    </search>
    <drilldown>
      <link
target="_blank">search?q=index%3Dweb%20clientip%3D$click.value$</link>
    </drilldown>
  </table>
</panel>
```

5) Tokens & Forms (fast primer)

• Set a token from an input: <input type="dropdown" token="st">…</input> → use as st

• Time picker tokens: $tspan.earliest$, $tspan.latest$

• Drilldown tokens: $click.name$, $click.value$, $row.fieldname$

6) Performance Playbook (Classic)

• Prefer base + post-process (one heavy search → many light transforms).

• Constrain with time pickers; avoid “All time”.

• Use Report Acceleration (for transforming searches) where applicable.

• Use Data Model Acceleration + tstats for CIM data at scale.

• Cache heavy panels via scheduled reports (then “Load from report”).

• Keep panel count reasonable; stagger refresh intervals (e.g., 60–300s).

• Avoid high-cardinality timechart by <field> unless filtered.

7) Auto-Refresh & Time Range (common asks)

```
<option name="refresh.display">progressbar</option>
<option name="refresh.auto.interval">120</option> <!-- seconds -->
```

Tip: Sync all panels with a single <input type="time"> and pass tokens to each <search>.

8) Prebuilt/Search-backed Panels

Save a Search as a Report → enable Schedule/Acceleration. In the dashboard panel, choose From Report so the panel loads results without running a live search.

9) Security, Permissions, Packaging

• Align permissions at dashboard, saved search, and macro levels.

• Put shared assets in an App; if using JS/CSS extensions, add to appserver/static and reference in Simple XML (check org security policies).

10) Troubleshooting Cheatsheet

• Panel blank/slow: check search job (limit time, use base/post-process).

• Token not applied: verify token names, quotes, and token scope.

• Drilldown broken: print $click.value$ test, confirm field exists in table.

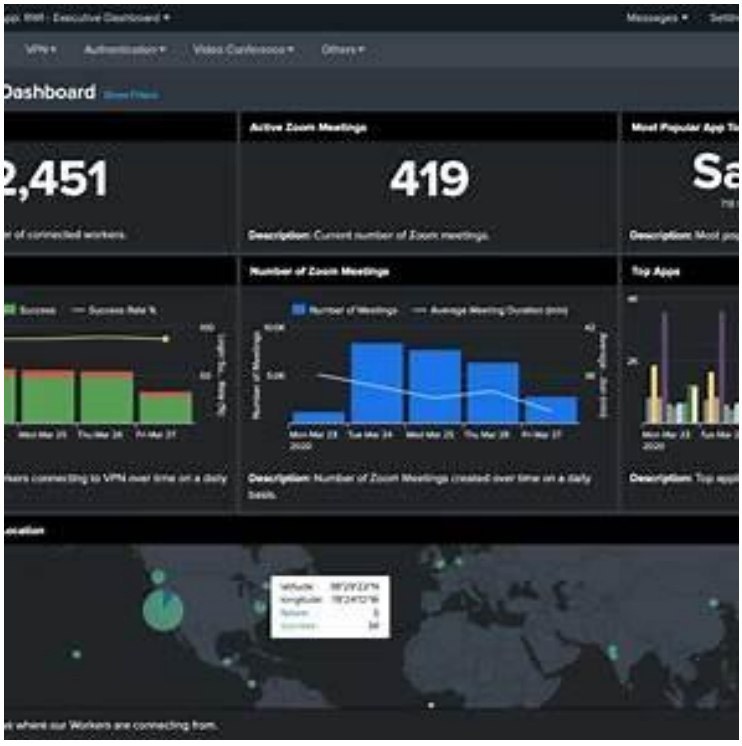
• Permissions errors: ensure user/app visibility to searches/macros/lookups.

• Auto-refresh not working: confirm per-panel interval and no browser auto-pause.

11) Migration Tips (Classic → Dashboard Studio)

Inventory panels + searches, note tokens/drilldowns. Rebuild visuals in Studio; keep searches the same. Replace advanced drilldowns with Studio's interactions + tokens. Validate role permissions and saved search dependencies.

12) Snapshot Checklist



13) Quick Demo Dataset (optional)

```
| makeresults count=500
| streamstats count
| eval _time=_time - (count*60)
| eval status=if(count%10<7,200,if(count%10<9,404,500))
| eval clientip="10.0.0.".(1+count%20)
| eval
uri_path="/app/".mvindex(split("home,login,cart,checkout,search,profile
",","), count%6)
| table _time status clientip uri_path
```

14) Handy SPL Snippets

Top errors by URL (last 4h):

```
index=web status>=400 earliest=-4h
| stats count by uri_path status
| sort - count
```

Golden 4xx/5xx trend:

```
index=web status>=400
| eval class=if(status>=500,"5xx","4xx")
| timechart count by class
```

15) Mini Flowchart (how a Classic dashboard works)