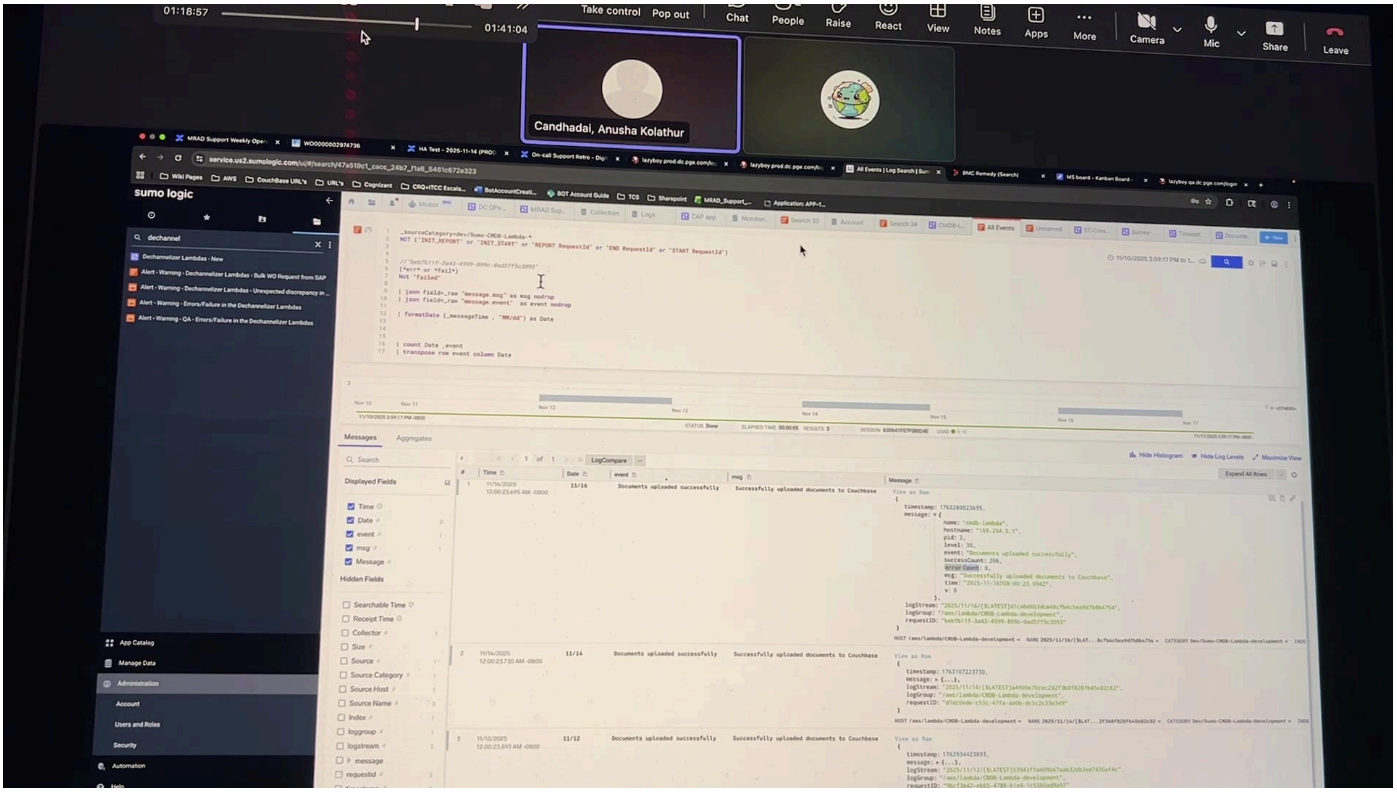
dash4 (cmdb lambda use case).mp4_44 | Summarize Videos, Audio, PDF & Websites

lilys.ai/digest/6962821/7341147



```
_sourceCategory=dev/Sumo-CMDB-Lambda-*
NOT ("INIT_REPORT" or "INIT_START" or "REPORT RequestId" or "END RequestId" or "START RequestId")


//"bebfb11f-3a43-4999-899c-0ad5ff5c3093"
(*err* or *fail*)
Not "failed"

| json field=_raw "message.msg" as msg nodrop
| json field=_raw "message.event"  as event nodrop

| formatDate (_messageTime , "MM/dd") as Date



| count Date ,event
| transpose row event column Date
```

Table of Contents

This session dives deep into **CMDB Lambda use cases** for practical log analysis and dashboard creation. You will learn specific techniques for **filtering error counts** versus success records within complex logs. Discover how to effectively structure queries to build dedicated **error summary** and **success panels** for clear operational visibility.

This summary details the process of analyzing CMDB Lambda logs for error and success tracking. The focus shifts to filtering logs and creating dedicated summary panels. [1]

Extracting capture image...

1. **Initial Goal**: The speaker intended to show the **error count** but noted it was covered under the **failed record count** [1].

    1. The speaker considered processing or rolling up the data further [3].

    2. They decided to temporarily leave the current configuration alone [5].

2. **Reviewing Existing Panel**: A request was made to review the existing panel for other available data [6].

    1. It was confirmed that a panel for **red count** already exists in daily execution [9].

    2. The speaker realized adding an **error count** panel was necessary [9].

    3. They clarified that the current panel shows the **count of records** at a number level [11].

Extracting capture image...

1. **Query Difficulty**: The speaker experienced difficulty interacting with the query interface [14].

    1. The system was described as consuming mental energy ("eats our brain") [19].

    2. The immediate goal was to filter logs specifically for **ERR** (error) and *not* **fail** [20].

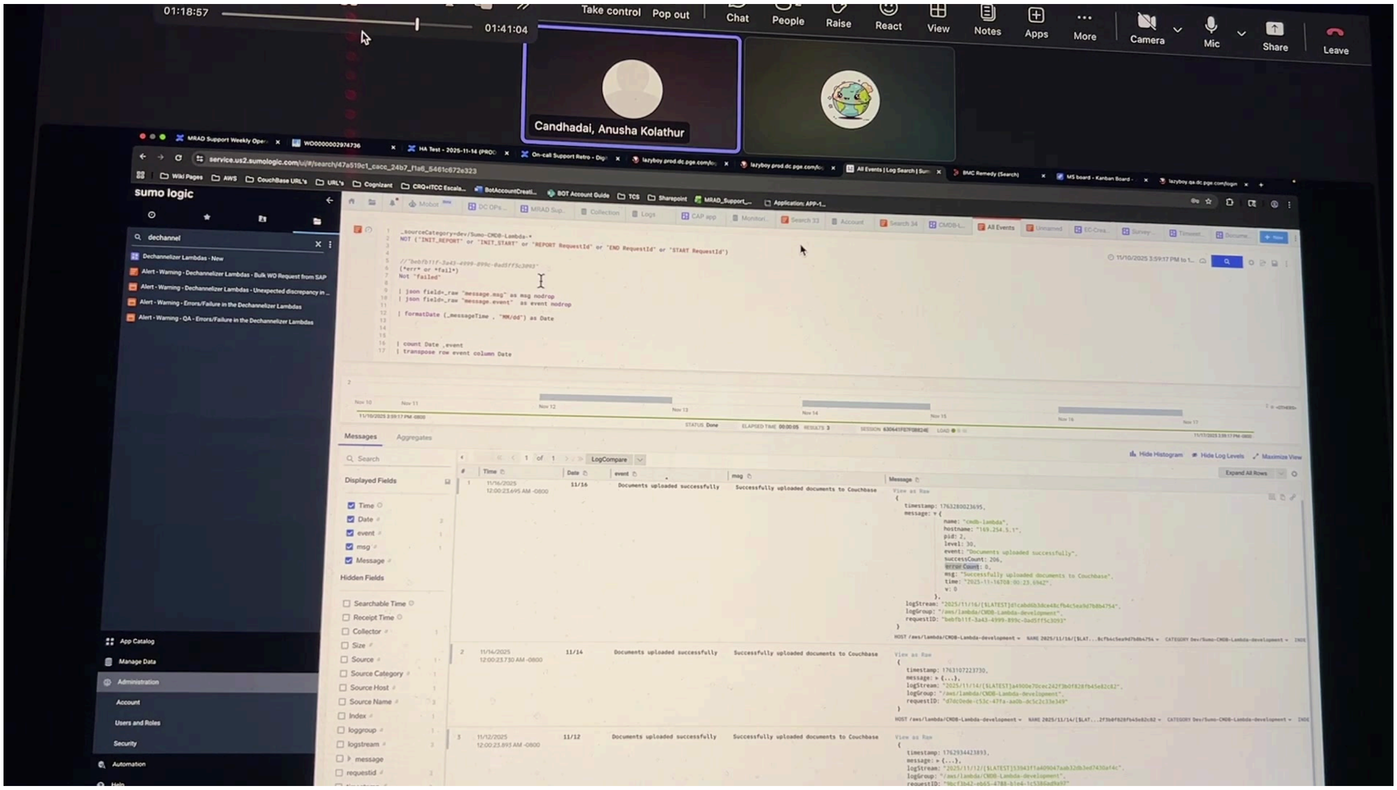
    3. The speaker wanted to confirm the next log after the current one being processed [25].

    4. A log showing **error count as zero** was observed [28].

    5. The current log processing was deemed sufficient for the table assembly [29].

    6. A decision was made to potentially rename the count as **error count** [32].

    7. Overall, the current status was considered covered and acceptable [33].



```
_sourceCategory=dev/Sumo-CMDB-Lambda-*
NOT ("INIT_REPORT" or "INIT_START" or "REPORT RequestId" or "END RequestId" or "START RequestId")


//"bebfb11f-3a43-4999-899c-0ad5ff5c3093"
(*err* or *fail*)
Not "failed"

| json field=_raw "message.msg" as msg nodrop
| json field=_raw "message.event"  as event nodrop

| formatDate (_messageTime , "MM/dd") as Date



| count Date ,event
| transpose row event column Date
```

#	Time	Date	event	msg
1	11/16/2025 12:00:23.695 AM -0800	11/16	Documents uploaded successfully	Successfully uploaded documents to Couchbase
2	11/14/2025 12:00:23.730 AM -0800	11/14	Documents uploaded successfully	Successfully uploaded documents to Couchbase
3	11/12/2025 12:00:23.893 AM -0800	11/12	Documents uploaded successfully	Successfully uploaded documents to Couchbase

```
{
  timestamp: 1763280023695,
  message: ▾ {
        name: "cmdb-lambda",
        hostname: "169.254.5.1",
        pid: 2,
        level: 30,
        event: "Documents uploaded successfully",
        successCount: 206,
        errorCount: 0,
        msg: "Successfully uploaded documents to Couchbase",
        time: "2025-11-16T08:00:23.694Z",
        v: 0
  },
  logStream: "2025/11/16/[$LATEST]d1cabd6b3dce48cfb4c5ea9d7b8b4754",
  logGroup: "/aws/lambda/CMDB-Lambda-development",
  requestID: "bebfb11f-3a43-4999-899c-0ad5ff5c3093"
}
```

1. **Filtering Goal**: The next step involved filtering out error counts and specific keywords [36].
    1. The speaker decided to temporarily only get the **event name** [39].
    2. Initial filtering showed **no matching records** for the current criteria [40].
    3. The best approach seemed to be filtering by **event kind** expression [43].
    4. Filtering by the keyword **failed** might miss actual failures [44].
    5. **New Strategy**: Filter using **upload complete** instead of **failed keyword** [45].
        1. This change helped resolve the immediate issue [47].
        2. The upload process itself mentions failure counts, confirming the strategy [48].
        3. Filtering by **failed** might lose sight of actual failures [49].
        4. Therefore, the log associated with **successfully uploaded** status was chosen [50].
        5. If uploaded successfully, the record should not contain an error log [52].
        6. Patterns with **failed** or **error** icons should be identifiable based on this [53].

Extracting capture image...

1. **Panel Creation**: The chosen log configuration would become the **error panel** [56].

    1. The speaker planned to let the query run and then add it to the dashboard [58].

    2. The new panel on the dashboard would be named **error summary** [60].

    3. The setup was considered almost complete [63].

Extracting capture image...

1. **Lambda Function Context**: The CMDB lambda fetches data from the database and places it in the approach base [65].

    1. The final message indicates whether the process was **successful or not** [66].

    2. The log being examined has a **state for success** [67].

    3. The message reads **completed successfully**, and the message content is identical [68].

    4. The speaker questioned making an *exclusive* **success panel** [69].

    5. Typically, data is shown in two ways: **success and failure** [69].

    6. The failure panel has no clear idea of how failure error logs appear [70].

    7. The final event considered still shows the **state as success** [72].

    8. Filtering where `state is equal to success` might remove all error records [73].

    9. Failures for the day would be filtered out if this strict success filter is applied [74].

    10. **Decision**: Since an error summary exists, it is better to make this panel purely the **success** panel [76].

Extracting capture image...

1. **Editing the Success Panel**: The process involves editing the query in the panel [78].

    1. The edit involves adding a condition where the **state is equal to success** [79].

    2. The panel name was updated to **daily execution successes** or **daily successes** [80].

    3. This resulted in two confirmed panels: **success** and **failure record count** [84].

2. **Preparing for Graphical Representation**: The next step is graphical representation [86].

    1. The initial attempt showed **No data to display** [87].

    2. The query needed to be re-executed [89].

    3. The speaker checked the **state** field again [91].

    4. **Root Cause**: The state was blindly assumed as success without passing the state parameter in the query [94].

    5. This caused the "nothing to display" result [96].

    6. The **state** parameter was corrected by copying and updating it [97].

    7. The state was set to **success**, and the update was confirmed [101].

    8. The next step is to show this data graphically [102].

    9. To graph, the data must first be **transposed** [104].

This section covers the necessary data manipulation before visualization. [105]

1. **Transposition Requirement**: Data must always be transposed before showing it in graphs [105].

    1. The **date field** should be placed on the rows for easier display [106].

    2. The speaker questioned if the date field was unique [108].

    3. The **failed records** field was considered for the column structure [109].

    4. The date field was renamed for clarity in the context of the day [111].

    5. A **duplicate** action was performed on the query/data structure [117].

    6. A concern arose about the capitalization of the **date** field [119].

    7. The query was confirmed to run successfully after these changes [122].

    8. The speaker paused to check something related to **Sherlock** [124].

Extracting capture image...

1. **Sherlock Mention**: Sherlock is currently a frequent topic of discussion [127].
    1. The speaker confirmed the current view was indeed a **graph** [130].
    2. They recalled a previous dashboard setup for a "Flight" related item [132].
    3. The process used there involved **time slice** [136].
    4. *Time slice* defines how data is segmented [137].
    5. Content in double curly braces `{}` represents a **parameter** [138].
    6. The current straight query does not need the parameter [139].
    7. The parameterization steps will be shown later [141].
    8. The speaker decided against using the date parameter for now [143].

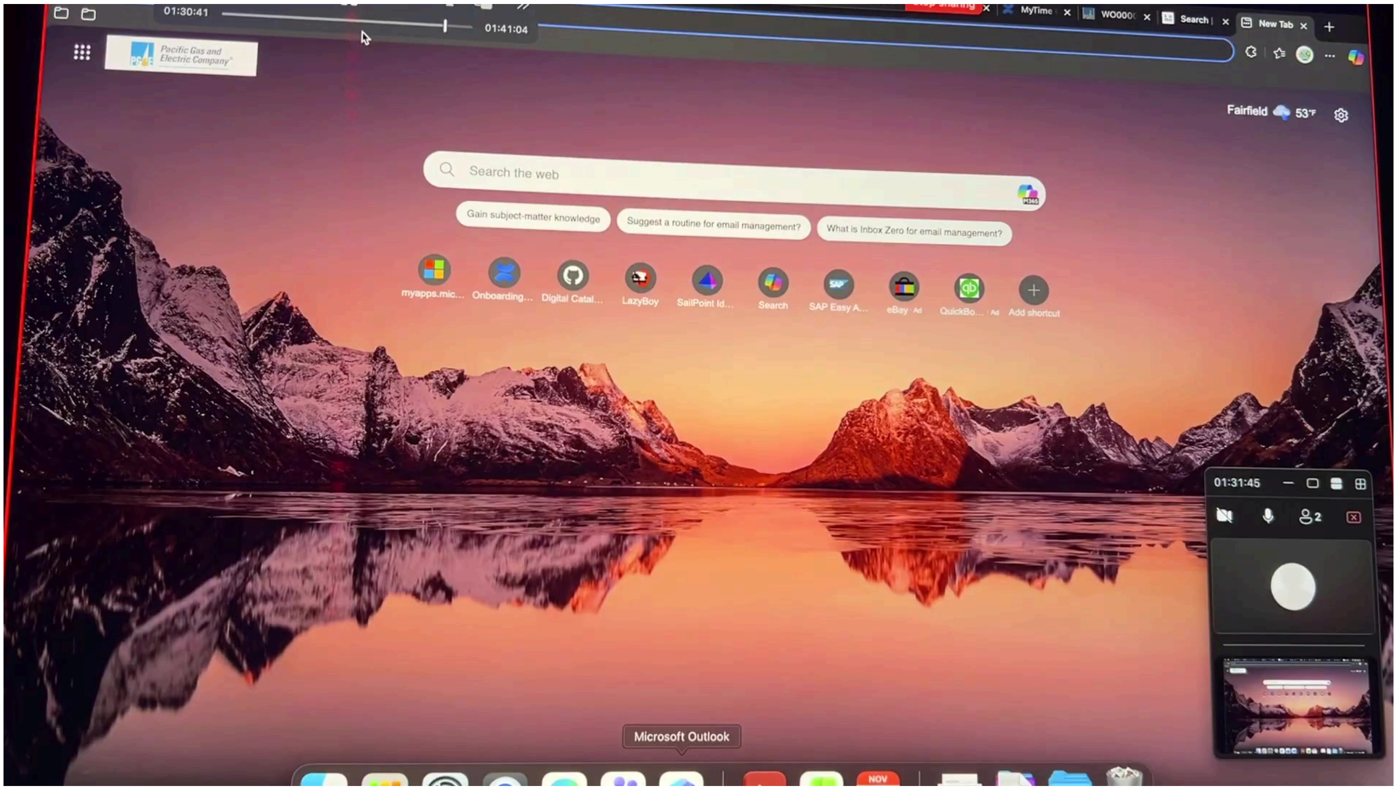
The conversation shifts to administrative tasks and a final technical adjustment for display consistency. [145]

1. **Communication Delay**: The speaker questioned if communications should be sent now or tomorrow [145].
    1. It was decided to handle **comms now** as it is important [148].
    2. They agreed to work on the technical aspects tomorrow [150].
    3. The current issue was deemed **not a blocker** [153].

Extracting capture image...

1. **Brief Interruption**: The speaker stepped away briefly [154].

    1. They mentioned an issue with an **eyelash getting stuck in the eye** [159].
    2. Hydration (drinking three bottles of water daily) was mentioned [161].
    3. The speaker clarified they were talking about **eyelids** getting stuck, not falling off [168].
    4. They apologized for the confusion regarding eyelids [173].
    5. The group agreed to proceed with the **comms** task [175].



1. **Inspect Checklist Review**: The speaker confirmed their inspect checklist text review was good [176].

    1. The speaker realized a fix for display issues [179].

    2. **Fix**: In **options**, choose **75% zoom** to standardize display [181].

        1. This prevents zooming in to type text from messing up the layout [182].

        2. The default text size was appearing too large otherwise [183].

        3. Setting it to **75%** standardizes the view [185].

    3. The speaker will now send out the **comms** notification [188].

        1. This is necessary because the task is **manual** [189].

    4. The final action was to proceed with sending the communications [191].