**Fetching & Filtering Logs**

Fetching and filtering logs helps retrieve specific log events from Dynatrace.

fetch logs

* Retrieves **all logs** from Dynatrace.
* No filters applied

📍 Fetch logs for the last 1 hour

fetch logs, from:now() - 1h

📍 Fetch error logs from the last 24 hours

fetch logs, from:now() - 24h

| filter level == "error"

📍 Filter logs by log level

fetch logs

| filter loglevel == "ERROR"

* Only retrieves logs where the **log level is "ERROR"**.

📍 Fetch logs containing a specific phrase

fetch logs

| filter matchesPhrase(content, "No files matching import")

* Retrieves logs containing the exact phrase **"No files matching import"** in their content.

📍 Fetch logs for a specific host

fetch logs

| filter dt.entity.host == "HOST-DC5BA1EFBA5CE3A6"

| filter matchesPhrase(content, "Channel Connectivity change to READY")

| fields timestamp, content

Retrieves logs **only from a specific host** and **containing a specific message**.

**Grouping & Summarizing Logs**

Summarizing logs allows aggregation, such as **counting occurrences** of logs by different attributes.

**📍 Count logs by log level**

fetch logs

| summarize count(), by:loglevel

Groups logs **by log level** and **counts occurrences**.

📍 Count logs by status (last 1 hour)

fetch logs, from:now() - 1h

| summarize count(), by:status

Groups logs **by status** and **counts them** within the last **1 hour**

📍 Count total logs & error logs (last 1 hour)

fetch logs, from:now() - 1h

| summarize Totalcount = count(), errortotal = countIf(status == "ERROR")

 Counts **all logs (Totalcount)**.

 Counts logs where **status is "ERROR" (errortotal)**.

📍 Calculate error percentage (last 1 hour)

fetch logs, from:now() - 1h

| summarize Totalcount = count(), errortotal = countIf(status == "ERROR")

| fieldsAdd errorpercent = (toDouble(errortotal) \* 100 / Totalcount)

 Counts **total logs** and **error logs**.

 Calculates **error percentage** using errorpercent = (error logs / total logs) \* 100.

**📍 Summarize logs by log source (Top 10)**

fetch logs

| filter loglevel == "ERROR"

| summarize by:{log.source}, count()

| sort `count()` desc

| limit 10

 Counts logs **by log source**.

 **Sorts** in **descending order**.

 Limits results to **top 10 log sources**.

📍 Show top 5 log sources by volume

fetch logs

| summarize by:{source = log.source}, volume = count()

| sort volume desc

| limit 5

**Time-Series Analysis**

Time-series helps **track log trends over time**.

**📍 Show logs over time, grouped by log level (excluding NONE)**

fetch logs

| filterOut loglevel == "NONE"

| makeTimeseries by:{loglevel}, interval:1h, count = count()

 Groups logs **by log level** over time.

 **Excludes** logs where **log level = NONE**.

 Aggregates logs in **hourly (1h) intervals**.

📍 Log volume trend over time (5-minute interval)

fetch logs

| makeTimeseries by:{loglevel}, interval:5m, count = count()

 Creates a **time-series** for logs.

 Groups logs **by log level**.

 Uses **5-minute intervals**.

📍 Show error logs over 7 days in 15-minute intervals

fetch logs, from:bin(now(), 7d)

| filter loglevel == "ERROR"

| makeTimeseries interval:15m, count = count()

 Retrieves **error logs from the past 7 days**.

 Groups them in **15-minute intervals**.

📍 HTTP 500 errors over the past 24 hours

fetch logs, from:now() - 24h

| filter matchesPhrase(content, "HTTP 500")

| summarize count(), by:{dt.entity.host}

| sort count() desc

 Filters logs containing **"HTTP 500"** errors in the last **24 hours**.

 Groups **by host** to see which servers are failing the most.

 Sorts in **descending order**.

📍 Maximum number of logs per hour (per log level & host)

fetch logs

| summarize by:{loglevel, dt.entity.host, \_time = bin(timestamp, 1h)}, max\_logs = count()

 Groups logs by log level, host, and hourly intervals.

 Finds maximum number of logs per hour.

**📍 Success rate based on response time < 75ms**

fetch logs

| parse content, “LD ‘ResponseTime[‘ INT:response\_time”

| summarize success = countIf(response\_time < 75), total = count()

| fieldsAdd successRate = (100 \* toDouble(success) / toDouble(total))

**📍 Count warning & error logs per host**

fetch logs

| filter in(loglevel, {"WARNING", "ERROR"})

| summarize count(), by:dt.entity.host

| sort count() desc

 Filters **WARNING & ERROR logs**.

 Groups **by host**.

 Counts logs per **host** and sorts in descending order.

**6️.Advanced Aggregation & Summarization**

**📍 Find the most frequent error messages in logs (Top 5)**

fetch logs

| filter loglevel == "ERROR"

| summarize count(), by:content

| sort count() desc

| limit 5

 Filters **only ERROR logs**.

 Groups logs **by content** (actual log messages).

 Sorts in **descending order** to find the most **frequent error messages**.

 Limits to **Top 5 errors**.

**4️⃣ Performance Analysis (Response Times)**

These queries focus on **response time analysis**.

📍 Count requests with response time > 1000ms

**fetch requests**

**| filter responsetime > 1000**

**| summarize count() by requestName, responsetime bin(responsetime, 100)**

**📍 Find percentiles for a request (GET /cart)**

fetch requests

| filter requestName == "GET /cart" and responsetime > 5000

| summarize percentiles(responsetime, 50, 90, 95, 99) by requestName

* Retrieves requests for "GET /cart" with response times above 5000ms.
* Calculates percentiles (50th, 90th, 95th, 99th percentiles).

📍 Average response time per API request

fetch requests

| summarize avg(responsetime), by:requestName

| sort avg(responsetime) desc

 **Groups API requests** by requestName.

 **Calculates the average** response time per API.

 **Sorts** in descending order (**slowest APIs first**).