

SSI v2 5.4 Kibana 3 cheatsheet

- [Links](#)
- [Loading an existing dashboard](#)
- [Deleting a dashboard](#)
- [Save a dashboard as something else](#)
- [Export your dashboard](#)
- [Import a dashboard](#)
- [Modify a panel](#)
- [Duplicate a panel](#)
- [Rearrange panels](#)
- [Local vs UTC time](#)
- [Pulling out data for a specific time period](#)
- [Pulling out data for multiple airlines](#)
- [How to get the FE and BE logs](#)
 - [Method 1 Use any logviewer that works, put the timestamp and CARF/Prefix from the dashboard](#)
 - [Method 2 Use the ALFHelper \(note uses the new ALF ..which is unstable \)](#)
 - [Method 3 Use Tampermonkey script](#)
- [Seeing available dashboards or Opening a dashboard](#)
- [Sharing a temporary view](#)
- [Queries](#)
- [Multiple query example - compare elapsed time of FM activity vs CM Get pax :](#)
- [Multiple queries - choosing what to chart](#)
- [Pull out multiple error messages using a wildcard](#)
- [Check if exists or does not exist](#)
- [Filter by Peak](#)
- [Choosing Query syntax as Lucene / REGEX / TopN](#)
- [Using topN queries](#)
- [Inspecting the query Kibana is building](#)
- [Fix Events Table not loading data issue](#)
- [Charting the Elapsed time](#)
- [A simple export](#)

Links

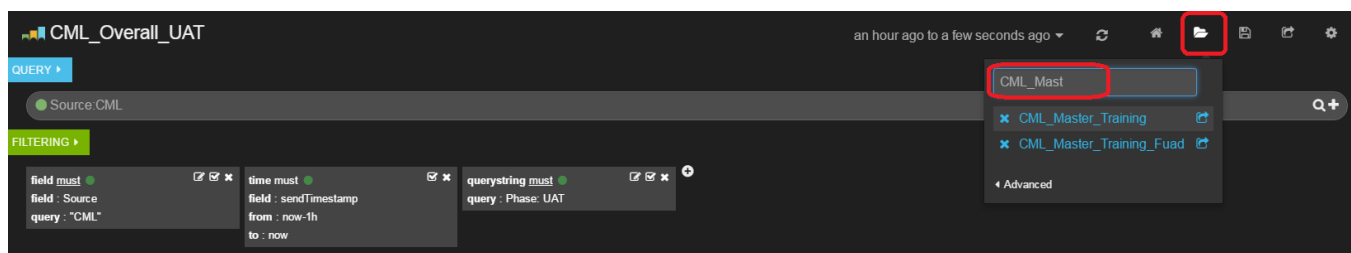
Video of Kibana session (unfortunately WEBEX failed to capture the audio) : https://amadeusworkplace-my.sharepoint.com/personal/amuni_amadeus_com/Documents/BOX/Kibana/Kibana_07FEB17.wrf

Links to all regions for ES/Kibana : [SSI v2 5.4 List of Elastic Search Regions](#), note we are using **Kibana 3**

List of critical DCS dashboards : [SSI v2 5.4 DCS Critical Dashboards PRD](#)

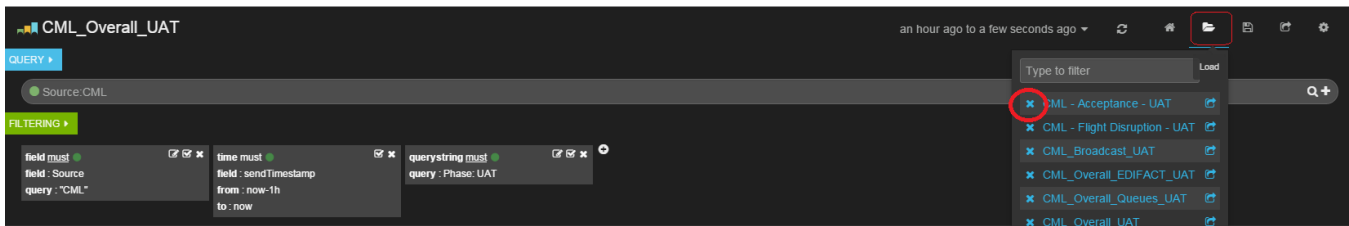
Loading an existing dashboard

Click on the Load icon, and start typing the name of the dashboard you want



Deleting a dashboard

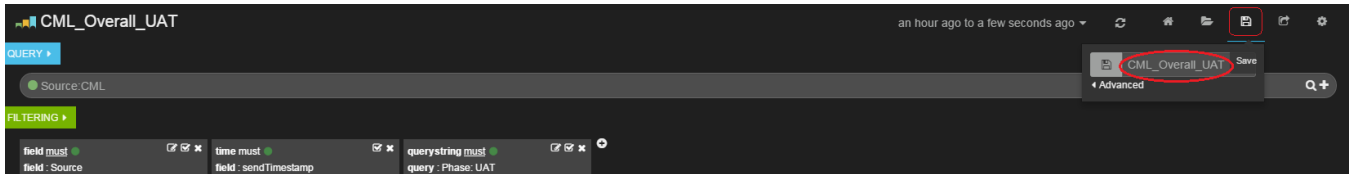
Click on the Load icon, look for the dashboard you want to delete, then click on the cross - X.



Save a dashboard as something else

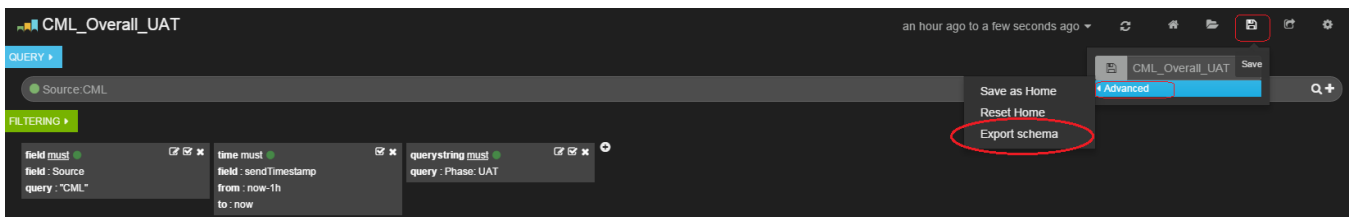
Click on the Save icon, then put the new name and click on the save icon to the **left** of the dashboard name.

There should be a green banner at the end then telling you your changes are saved.



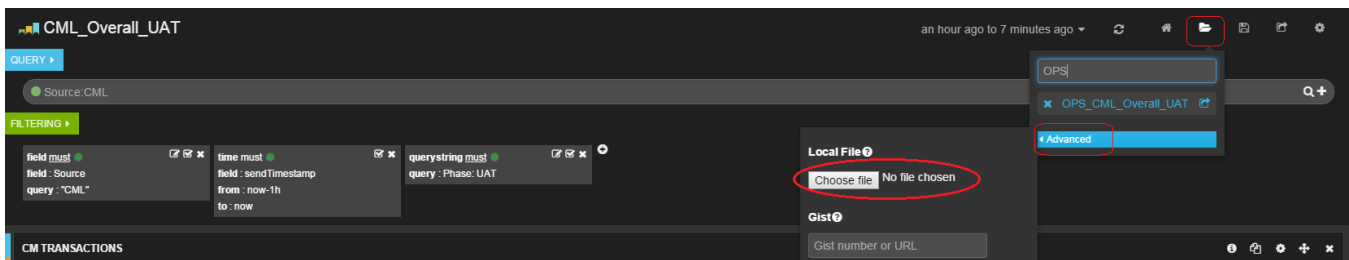
Export your dashboard

Click on the Save icon, then on Advanced, then select Export schema



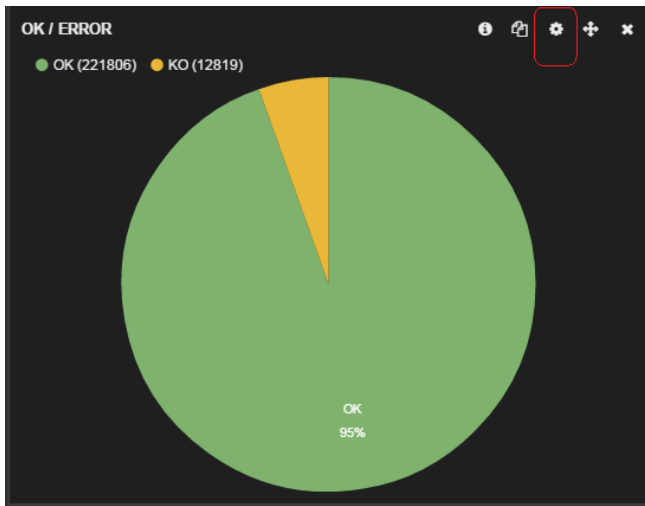
Import a dashboard

Click on the Load icon, then on Advanced, then select Local file and supply your local Json dashboard.



Modify a panel

Click on the Configure icon and then change any aspect you want - the title, the display etc. There are 3 panels.



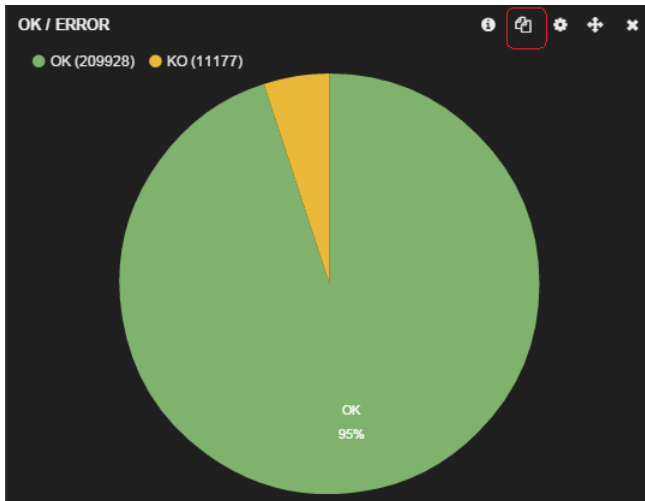
General Panel Queries

Stable // Displays the results of an elasticsearch facet as a pie chart, bar chart, or a table

Title: OK / Error Span: 4 Editable: ☒ Inspect: ☒

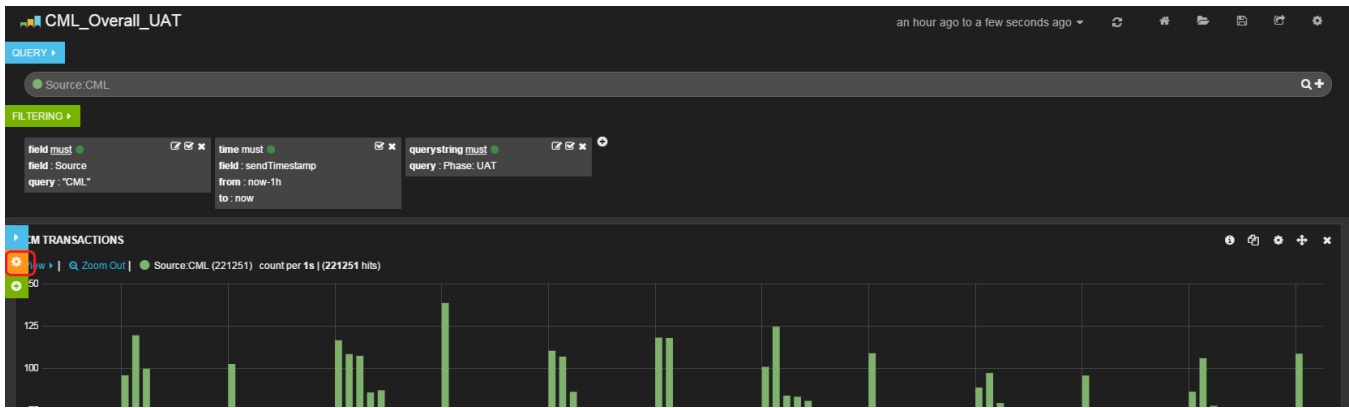
Duplicate a panel

Click on the Duplicate icon, then change the elements in the new panel item created on your dashboard via the Configure icon.



Rearrange panels

Click on the Configure icon near the Histogram, then use the Panels tab to rearrange.

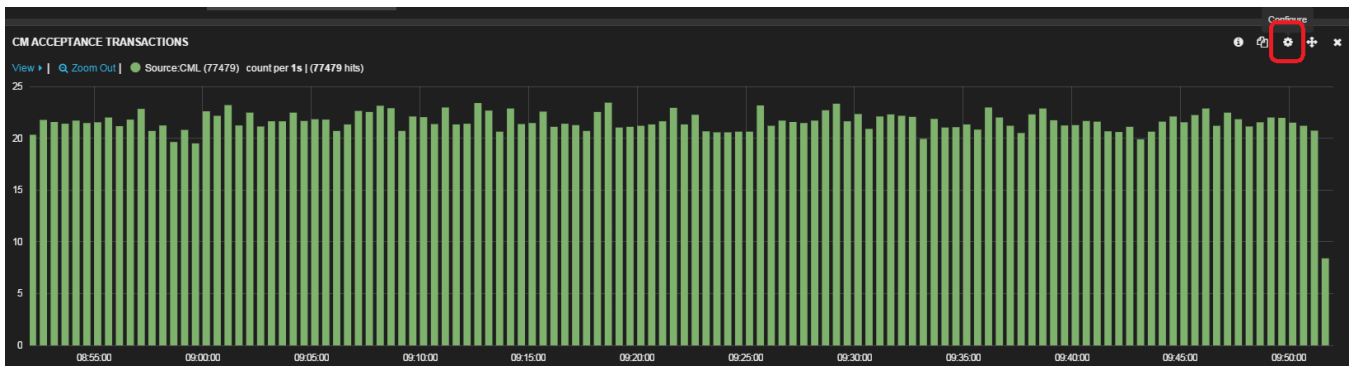


Panels							Row Settings	
Title	Type	Span (35/12)	Delete	Move		Hide		
CM Transactions	histogram	12	×		↓	■		
Services	terms	4	×	↑	↓	■		
Peak	terms	4	×	↑	↓	■		
OK / Error	terms	4	×	↑	↓	■		
Errors	terms	4	×	↑	↓	■		
Channel	terms	3	×	↑	↓	■		
Service Helper	text	4	×	↑		■		

Local vs UTC time

You can configure the histogram to show items in local or UTC. **The recommendation is to always have your histogram in UTC,**

Click the Configure icon near the top right of the histogram.

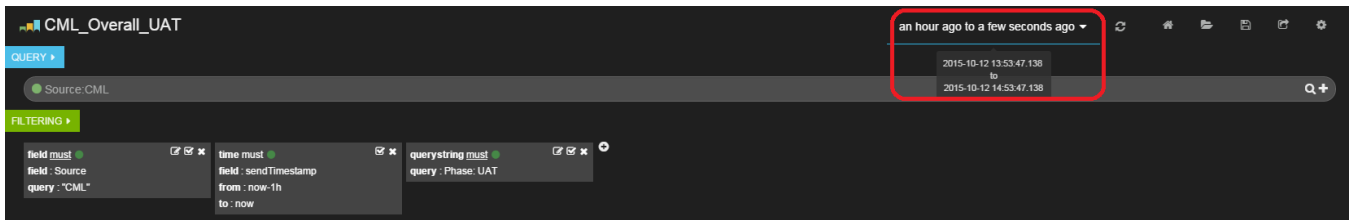


Go to Panel, and choose UTC or Browser as needed. Choosing Browser will give you local time.

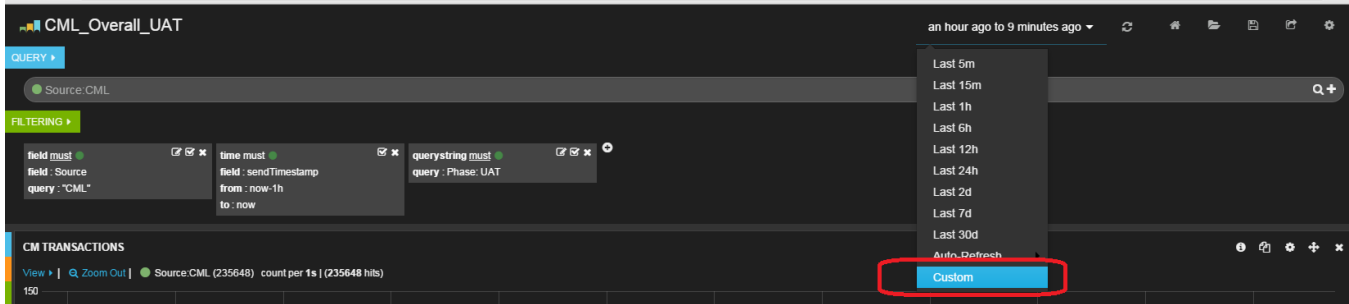
General				Histogram Settings	
Panel					
Values		Transform Series		Time Options	
Chart value	Seconds	Derivative	Zero fill	Time Field	Time correction
count	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	sendTimestamp	browser
					browser
					utc
				Auto-interval	Resolution
				<input checked="" type="checkbox"/>	100
				Save Cancel	

Pulling out data for a specific time period

Click on the arrow in the time period on the top right of your dashboard



Choose Custom



Enter the date and time you want and click on Apply

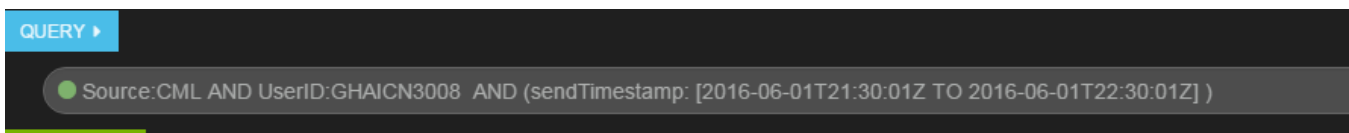


Note your input is Local time, and then the subsequent display in the Histogram is then converted to UTC (as we always choose UTC and not browser time in our dashboards).

Unfortunately Kibana doesnt let us choose an input in UTC for the time input.

Noted there is also a way to specify an explicit time input in the Query line, if needed

e.g. Source:CML AND Host:* AND sendTimestamp:[2016-06-01T21:30:01Z TO 2016-06-01T22:30:01Z]

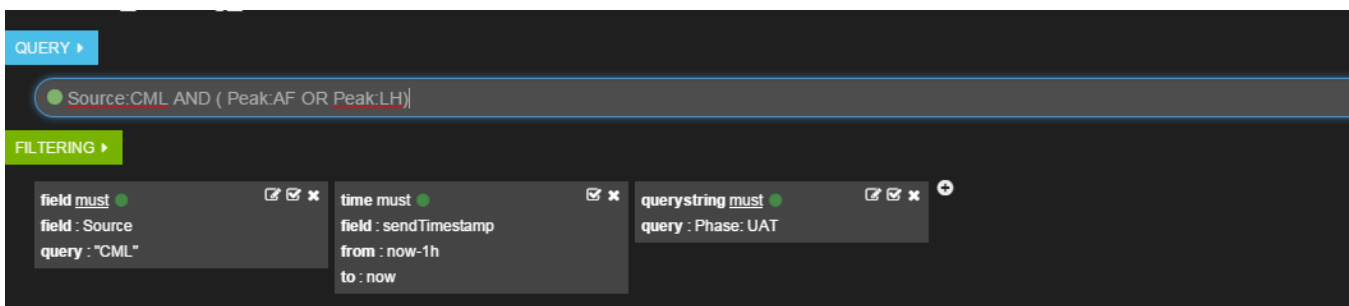


Pulling out data for multiple airlines

Here i am using the Peak as an example and pulling out data for LH and AF

Method 1

The easiest way will be to perform a query (refer the queries section for more details)



Method 2

Filter for the first Airline via the Piechart as usual, and change the filter to "Either"

terms

either

field

Peak

value

AF

Save

Apply

Click on the Piechart again, that will add a copy of the existing filter.

terms

either

field

Peak

value

AF

terms

must

field

Peak

value

AF

Now Edit the new filter - put "Either" and change the value to LH

querystring

must

query

Phase: UAT

terms

either

field

Peak

value

AF

terms

either

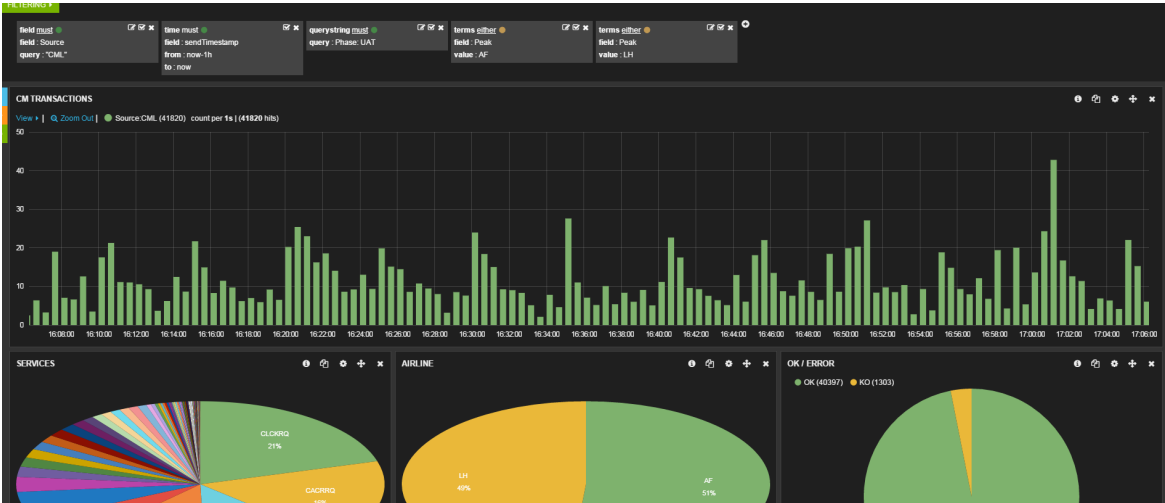
field

Peak

value

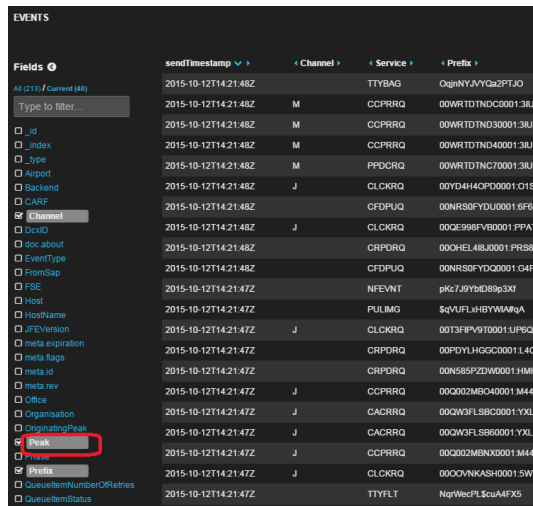
LH

Click on apply and you will have the data filtered for LH and AF



Method 3 (this only works when you add a filter via the Events table)

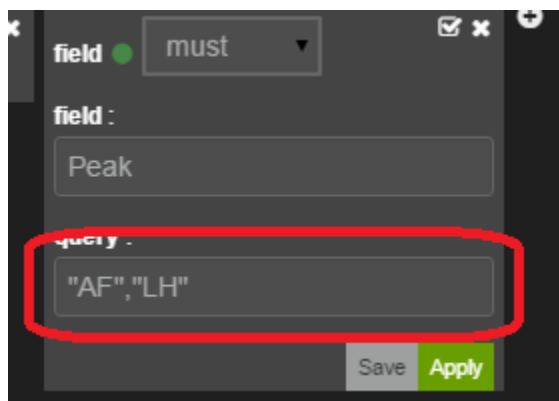
Go to the Events table and click on the field Peak



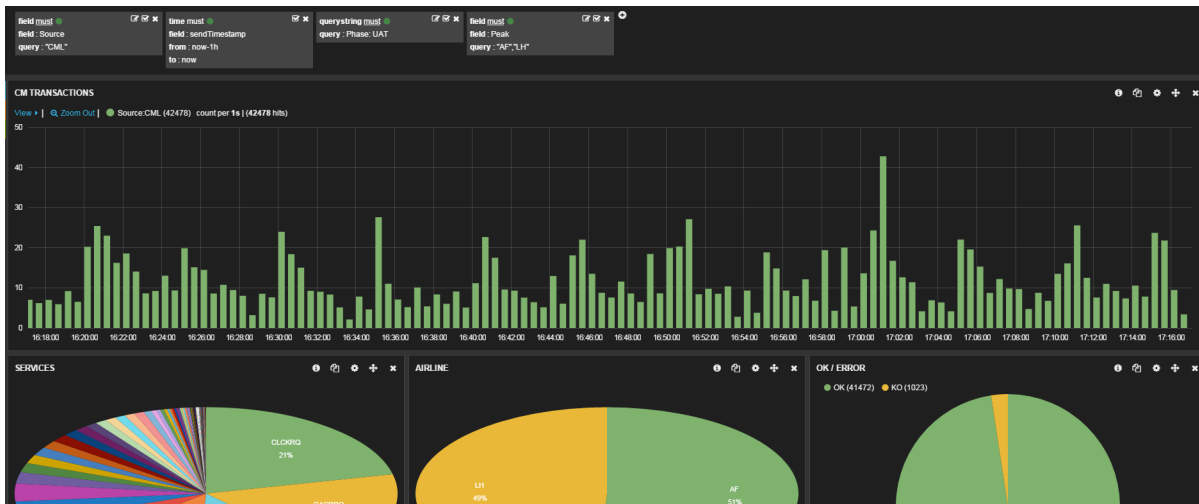
Click on the magnifying glass icon - this will add a filter for "AF" Peak **at the top of your dashboard**

	Value	Action	Count / 500 events
1.	MK		142
2.	AF		134
3.	6X		24
4.	LH		21
5.	BA		20
6.	AV		19
7.	KI		14

Go to the top of your dashboard, find this Peak filter, and edit to add the other airline too, separated by a comma



Click on apply and you will now have the data filtered for AF and LH



How to get the FE and BE logs

Method 1 Use any logviewer that works, put the timestamp and CARF/Prefix from the dashboard

We do not have an API from ALF which lets us build the logsearch URL on the fly (they need us to send the whole data in the payload itself, which is a lot).

Here is how you can get the logs yourself for any item you see on the dashboard.

We will get the logs for CLCKRQ as an example

1) Click on a row you are interested in to expand it. .

Fields	sendTimestamp	Channel	Service	Prefix	TransactionStatus
All (233) / Current (49)	2015-10-26T10:16:31Z		BDCQUE	5qQSYrj73O8H3Am	OK
Type to filter...	2015-10-26T10:16:31Z	J	CLCKRQ	00W7CVBGW50001:8PTA0KAXWBIZ1UA25RTAP3LY:1-2	OK
<input type="checkbox"/> _id	2015-10-26T10:16:31Z	J	CLCKRQ	00W7CVBGW30001:8PTA0KAXWBIZ1UA25RTAP3LY:1-1	OK
<input type="checkbox"/> _index	2015-10-26T10:16:31Z		SYNCUS	\$KqPYEx#ISWc1ba#	OK
<input type="checkbox"/> _type	2015-10-26T10:16:31Z	J	PSPIUQ	003ODU5KW0001:CRQM1DP9W\$.JY#R6#WYAEPD4H:1	OK
<input type="checkbox"/> Airport	2015-10-26T10:16:30Z		CRPDRQ	001H4YJ6W40001:CDKX1ZV1OFU2PRI9QYEOH5#91:6-1	OK
<input type="checkbox"/> ArrivalDateTime					

2) Use the relevant data to get either the FE log or the BE log

2015-10-26T10:16:31Z	J	CLCKRQ	00W7CVBGW50001:8PTA0KAXWBIZ1UA25RTAP3LY:1-2	OK
View: Table / JSON / Raw				
Field	Action	Value		
Backend	🔍	LCK_LCK_CMLU		
CARF	🔍	00010030DU5KWU	The CARF is the value in the UNH. This helps you find the Front end logs	
Channel	🔍	J		
DocID	🔍	8PTA0KAXWBIZ1UA25RTAP3LY		
EventType	🔍	CLCKRQ		
FromSap	🔍	952_RDIJ		
Host	🔍	gbeV664		
HostName	🔍	964_PK2_CMLU		
Peak	🔍	LH		
Phase	🔍	UAT		
Prefix	🔍	00W7CVBGW50001:8PTA0KAXWBIZ1UA25RTAP3LY:1-2	This is the PREFIX - i.e. the Covid + Doc, visible in the backend logs	
Service	🔍	CLCKRQ		
Source	🔍	CML		
TransactionID	🔍	00W7CVBGW5	The TransactionID is the OriginCovid from the UNB segment.	
TransactionStatus	🔍	OK		
Version	🔍	1		
_id	🔍	lgsu802_140281222317824_1445854591_eKE3HCY9QiaTBagSODUPxe		
_index	🔍	turnon26102015		
_type	🔍	dcslDocument		
meta.expiration	🔍	0		
meta.flags	🔍	0		
meta.id	🔍	lgsu802_140281222317824_1445854591_eKE3HCY9QiaTBagSODUPxe		
meta.rev	🔍	1-00018c5840c958860000000000000000		
sendTimestamp	🔍	2015-10-26T10:16:31Z	The timestamp in UTC for the log retrieval	

3) e.g. using the CARF to get the FE log

Scope /

Phase

Applications

File types Log file names

Date Info: message-retention / truncation

Time (GMT) now 2

Pattern

The sendTimestamp and CARF values are used to search the FE logs

This results in the following, and you can get the Backend log from the FE log

1 / FE	2015/10/26 10:16:30.967733	obevt964	FE1C4VZ002POHQ5-983090	PSPIUQ:15:1:1A	ConvId: 00300U5KWU0001	Size: 0 kbytes
2 / FE	2015/10/26 10:16:30.988515	obevt964	FE1C4VZ002POHQ5-983090	CLCKRQ:13:1:1A	ConvId: 00W7CVBGW30001	Size: 0 kbytes
3 / FE	2015/10/26 10:16:31.004590	obevt964	FE1C4VZ002POHQ5-983090	CLCKRR:13:1:1A	ConvId: 00U6XTEQ00001	Size: 0 kbytes
4 / FE	2015/10/26 10:16:31.039496	obevt964	FE1C4VZ002POHQ5-983090	CLCKRQ:13:1:1A	ConvId: 00W7CVBGW50001	Size: 0 kbytes
5 / FE	2015/10/26 10:16:31.055666	obevt964	FE1C4VZ002POHQ5-983090	CLCKRR:13:1:1A	ConvId: 00U6JDUHC0001	Size: 0 kbytes
6 / FE	2015/10/26 10:16:31.666521	obevt964	FE1C4VZ002POHQ5-983090	CACRRQ:14:1:1A	ConvId: 00W7CVBGW90001	Size: 0 kbytes
7 / FE	2015/10/26 10:16:31.680692	obevt964	FE1C4VZ002POHQ5-983090	CACRRR:14:1:1A	ConvId: 00UFDX35AF0001	Size: 0 kbytes
8 / FE	2015/10/26 10:16:31.685516	obevt964	FE1C4VZ002POHQ5-983090	CACRRQ:14:1:1A	ConvId: 00W7CVBGW0001	Size: 0 kbytes
9 / FE	2015/10/26 10:16:31.705797	obevt964	FE1C4VZ002POHQ5-983090	CACRRR:14:1:1A	ConvId: 00UFDX35AH0001	Size: 0 kbytes
10 / FE	2015/10/26 10:16:31.736317	obevt964	FE1C4VZ002POHQ5-983090	CACRRQ:14:1:1A	ConvId: 00W7CVBGW0001	Size: 0 kbytes
11 / FE	2015/10/26 10:16:31.755528	obevt964	FE1C4VZ002POHQ5-983090	CACRRR:14:1:1A	ConvId: 00UFDX35AJ0001	Size: 0 kbytes
12 / FE	2015/10/26 10:16:31.758312	obevt964	FE1C4VZ002POHQ5-983090	CACRRQ:14:1:1A	ConvId: 00W7CVBGW30001	Size: 0 kbytes
13 / FE	2015/10/26 10:16:31.779442	obevt964	FE1C4VZ002POHQ5-983090	CACRRR:14:1:1A	ConvId: 00UFDX35AL0001	Size: 0 kbytes
14 / FE	2015/10/26 10:16:31.861671	obevt964	FE1C4VZ002POHQ5-983090	CLCKRQ:13:1:1A	ConvId: 00W7CVBGW0001	Size: 0 kbytes
15 / FE	2015/10/26 10:16:31.878340	obevt964	FE1C4VZ002POHQ5-983090	CLCKRR:13:1:1A	ConvId: 00U6XTEQ0001	Size: 0 kbytes
16 / FE	2015/10/26 10:16:32.036726	obevt964	FE1C4VZ002POHQ5-983090	PSPIUR:15:1:1A	ConvId: 00W7CVBGW10001	Size: 0 kbytes

4) e.g. using the Prefix to get the BE log (or all logs - just choose ALL instead of BE)

Scope /

Phase

Applications

File types Log file names

Date Info: message-retention / truncation

Time (GMT) now 2

Pattern

The sendTimestamp and Prefix values are used to search the BE logs

This results in the following

1 / BE	2015/10/26 10:16:31.041490	obevt964	LCK_LCK_CMLU#1-995276	YDW INFO	<ServiceManager.cpp#1274 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] preinvoke succeeded
2 / BE	2015/10/26 10:16:31.041514	obevt964	LCK_LCK_CMLU#1-995276	YDW INFO	<ExecutionTimeManager.cpp#433 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] Starting execution timer (ID=129207)
3 / BE	2015/10/26 10:16:31.041558	obevt964	LCK_LCK_CMLU#1-995276	DB TST	<OTFService.cpp#97 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] OTFDBCallbackService invoked
4 / BE	2015/10/26 10:16:31.042704	obevt964	LCK_LCK_CMLU#1-995276		[PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] PeakToken 'AIRIT,LH' leads to schema suffix 'lh' within application peak 'cmlu_peak2'
5 / BE	2015/10/26 10:16:31.043540	obevt964	LCK_LCK_CMLU#1-995276	APP INFO	<EntObeContext.cpp#74 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] >>>> constructing new EntObeContext
6 / BE	2015/10/26 10:16:31.043769	obevt964	LCK_LCK_CMLU#1-995276	APP INFO	<EntContext.cpp#76 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] >>>> constructing new EntContext
7 / BE	2015/10/26 10:16:31.043797	obevt964	LCK_LCK_CMLU#1-995276	APP TST	<BasTransactionAbstract.cpp#23 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] listeners cleared
8 / BE	2015/10/26 10:16:31.044045	obevt964	LCK_LCK_CMLU#1-995276	APP TST	<SvcUtils.cpp#92 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] decoding EDIFACT message <UNB\xIdIATB\xId1A0BE\xId1
9 / BE	2015/10/26 10:16:31.044221	obevt964	LCK_LCK_CMLU#1-995276	APP INFO	<NGDCallbackService.cpp#328 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] Package: NCMLCKU
10 / BE	2015/10/26 10:16:31.044244	obevt964	LCK_LCK_CMLU#1-995276	APP INFO	<NGDCallbackAbstract.i#54 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] <<<<< New Entry Received, OTF ServiceObject: I
11 / BE	2015/10/26 10:16:31.044343	obevt964	LCK_LCK_CMLU#1-995276	APP INFO	<EntAbstract.cpp#86 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] processing CLCKRQ associated to N3LCK14SvcRequestLoc
12 / BE	2015/10/26 10:16:31.044438	obevt964	LCK_LCK_CMLU#1-995276	APP INFO	<EntAbstractHelper.i#24 TID#5> [PFX: 3047CVBGH58001:8PTA0KAXMBI711JAZ5RTAP3LY:1-2] Inbound message decoded into input Svi

Method 2 Use the ALFHelper (note uses the new ALF ..which is unstable)

1. Click on a row you are interested in to expand it
2. Copy the JSON
3. Put it in <http://fumatv01.os.amadeus.net:9090/ALFHelper/>
4. When you click on View Front End Log or View Backend Log, it builds an input to the new ALF. The search response then depends on whether the new ALF works or not..

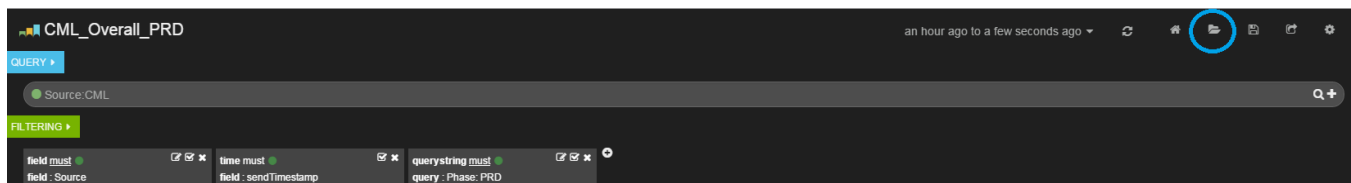
Method 3 Use Tampermonkey script

You may want to consider [click, select, right click + click](#) approach using a custom [Tampermonkey](#) script "Kibana ALF search".

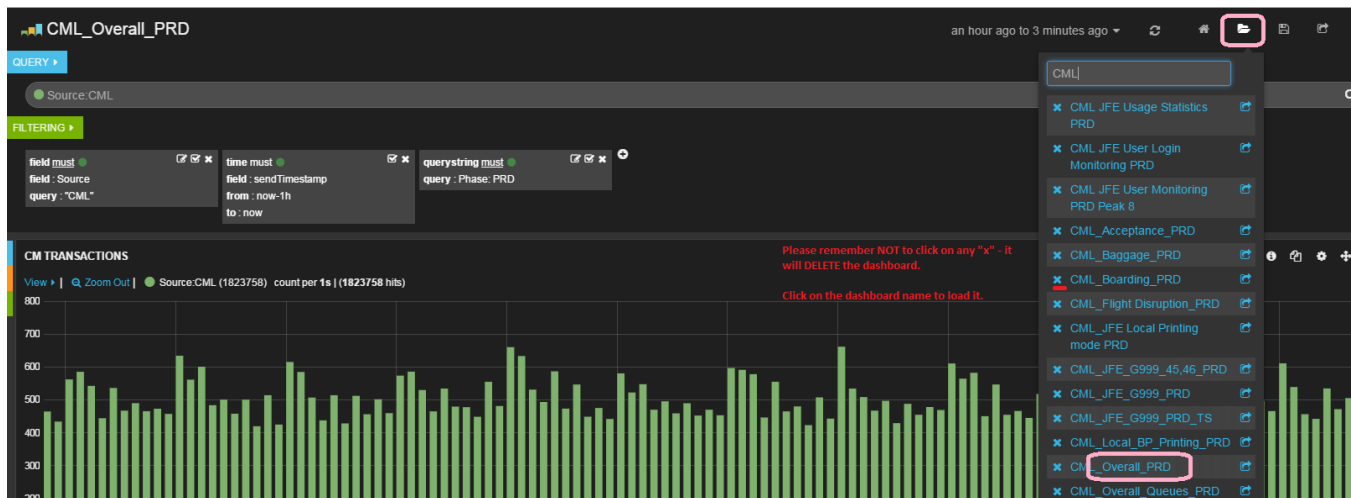
1. Go to a Kibana dashboard and find a table with CARF's you want to search
2. click on the row you want to search: you will see "View: Table / JSON / Raw" - click on **Raw** which will show you a raw JSON data
3. select all the JSON data
4. right-click and you should see Tampermonkey sub-menu - select "Kibana - search by CARF in ALFv3" from there
5. this will open a separate window with ALFv3 page with pre-filled values: you can adjust them (perhaps the time span) and then click Submit

Seeing available dashboards or Opening a dashboard

Use the Load icon, then type the name. CML dashboards start with CML_ , FML ones with FML_ , all as per naming conventions here : [SSI v2 5.4 Architecture and Basics#Dashboardnamingconventions](#)



Click on the dashboard name and NOT on the "x" to the left of the name (clicking on the "x" deletes the dashboard)



Sharing a temporary view

Say you have an incident, and you filter for an error / some airlines.

You can share this temporary view that you are seeing with everyone, without saving it and impacting the existing basic dashboard. It will be available even after the incident.

CML_Overall_PRD

6 hours ago to a few seconds ago

QUERY

Source: CML

FILTERING

field must

field : Source

query : "CML"

time must

field : sendTimestamp

from : now-6h

to : now

querystring must

query : Phase: PRD

terms must

field : TransactionStatus

value : KO

terms must

field : TransactionErrors.Errors.Error

value : 727 - INTERNAL ERROR: Failed to load the values. Could not find data. - Mdw exception received in TaskResolveDocuments.preTask - Business exception received in TDSDatabase.execute

Temporary filters have been applied on the Overall view, for a specific Incident

CML_Overall_PRD

shareable link

Share this dashboard with this URL

<http://fomap303.os.amaheus.net:8080/kibana-dev/#dashboard/temp/of2jRWexJSxCrdRjCY5eIA>

Copy this URL to share your view, with all your filters, without having to save and affect the main dashboard

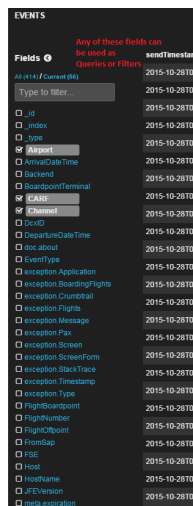
Close

Queries are a lot more powerful and are the ones responsible for retrieving the data you see. Filters are applied on top of queries.

Its good to have a performant query, which is specific and targetted.

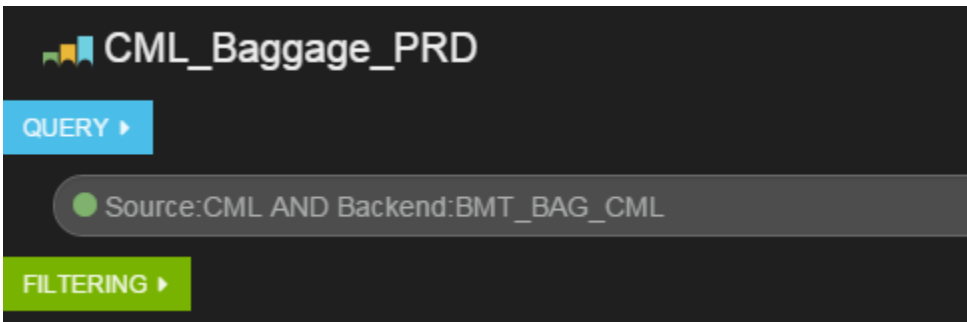
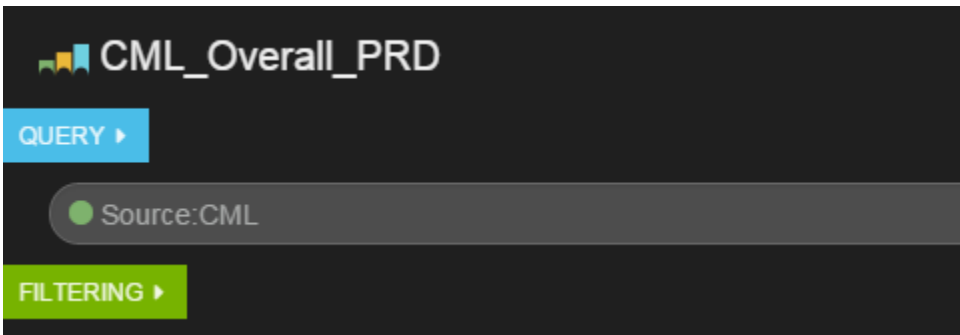
Any fields that you see in the logged JSON can be used in your Queries (refer the Event table visible at the bottom of each Dashboard)

You can have multiple queries, and compare two different transactions all in the same dashboard.



A dashboard for an area like Baggage, which wants to pull out all the Baggage related transactions can have a more targetted query by saying **Source: CML AND Backend:BMT_BAG_CML**

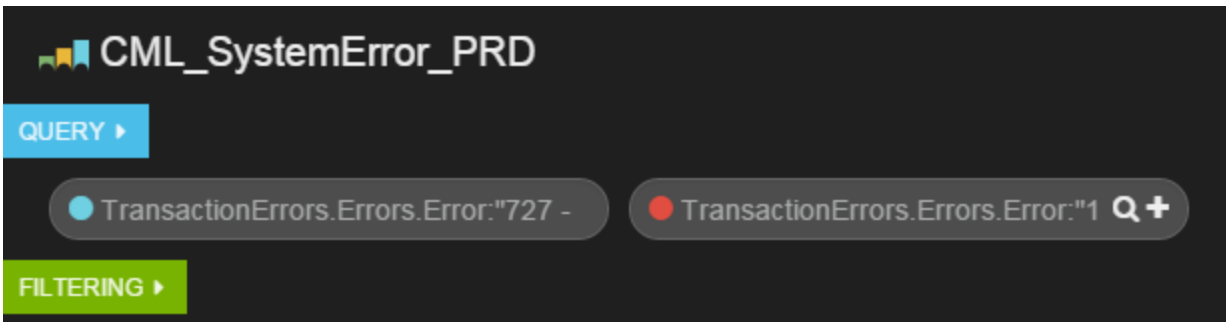
For more details on Query Syntax - please refer : <https://www.elastic.co/guide/en/kibana/3.0/queries.html> and also <https://www.mjt.me.uk/posts/kibana-101/>



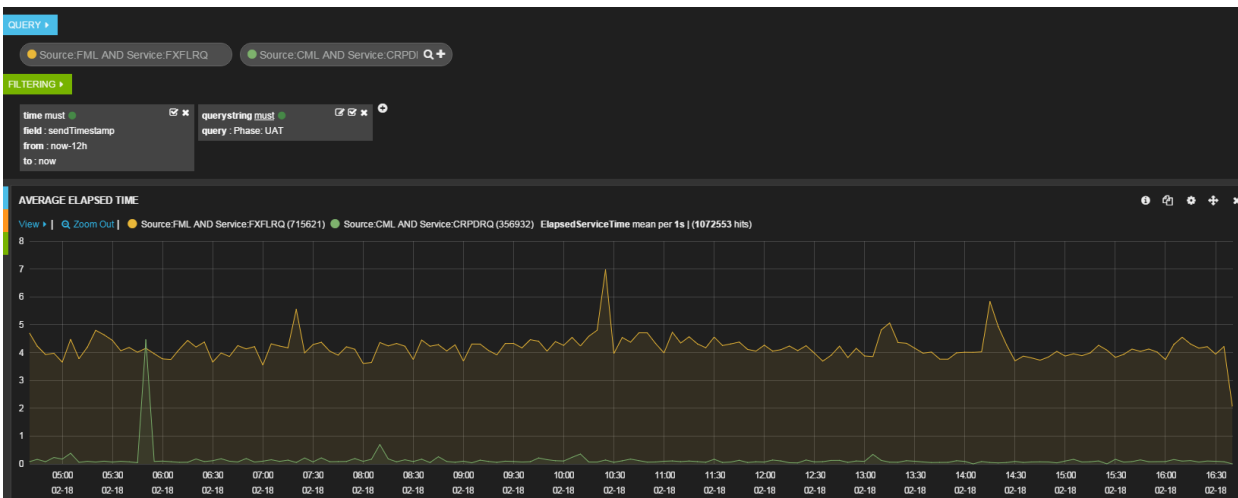
Multiple queries

Multiple queries are allowed too - but it is upto you to decide if your dashboard really needs to fire multiple queries.

Please aim to have queries targeting specific fields so that your dashboard is performant.



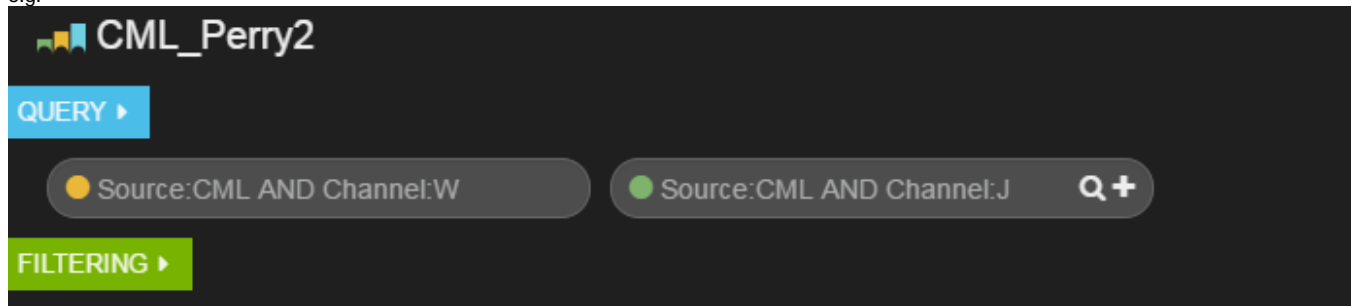
Multiple query example - compare elapsed time of FM activity vs CM Get pax :



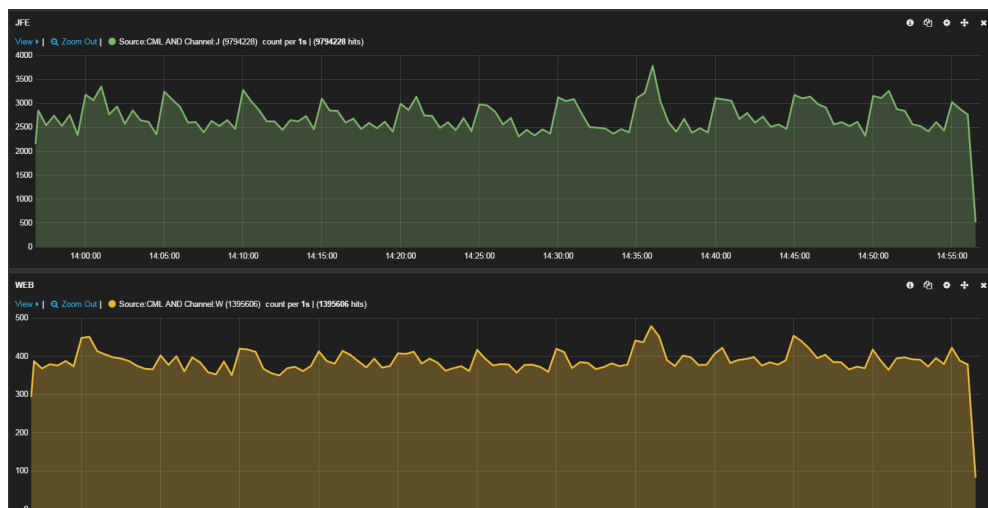
Multiple queries - choosing what to chart

When your dashboard has multiple queries, you can choose a specific one per chart if you want. By default all queries are selected.

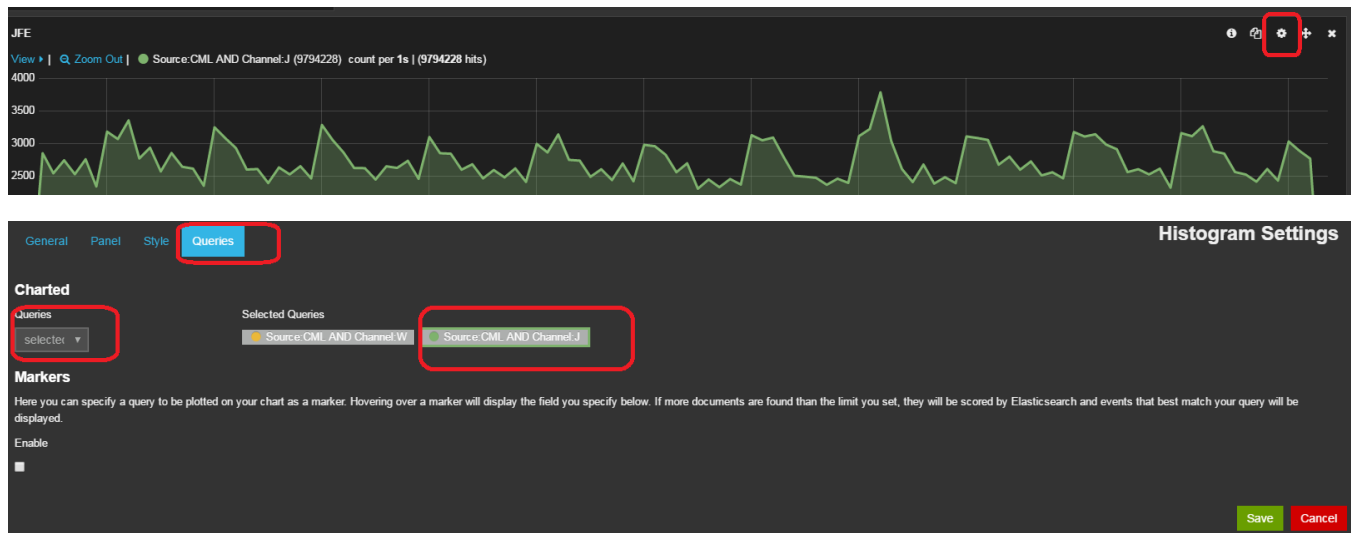
e.g.



Histogram 1 - charts only query 1 and Histogram 2 charts only query 2



Go to settings for each histogram, and just select the queries you want to chart. Save, then Save the dashboard.

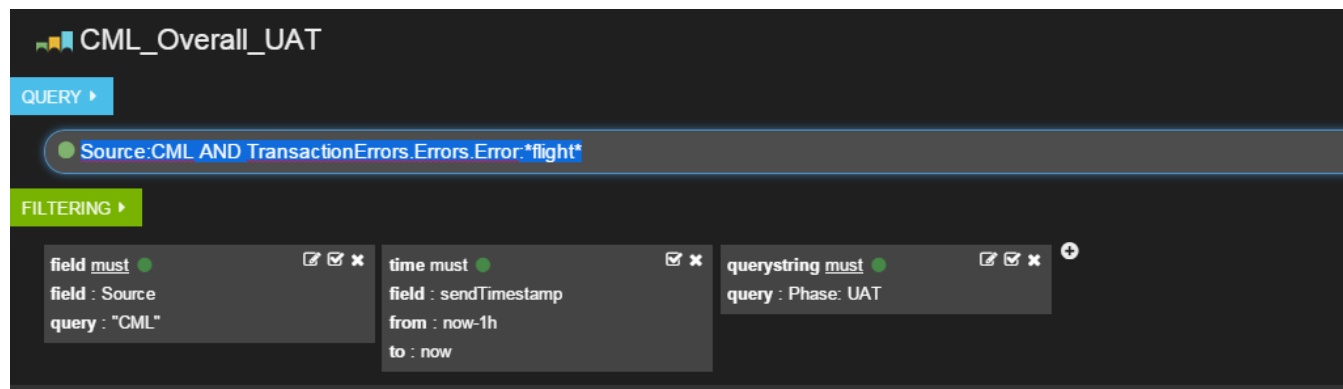


Pull out multiple error messages using a wildcard

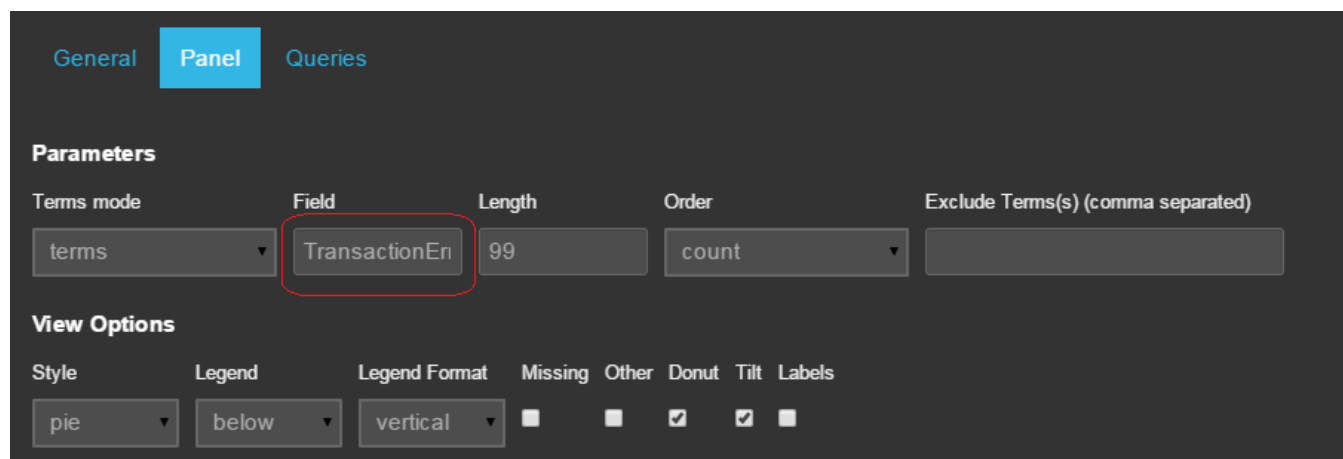
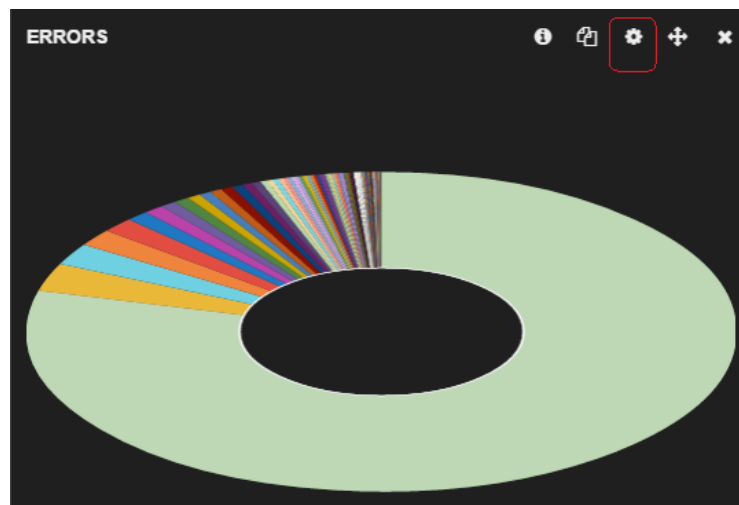
This is indeed possible via a Query.

e.g. see the input for showing all the errors with "flight" in the error message text for CML

Source:CML AND TransactionErrors.Errors.Error:*flight*



Note to know the field name to pull out for errors, i clicked the configure icon on the Errors chart, went to the Panel tab, and copied the value in "Field"



For more help on wildcards : https://www.elastic.co/guide/en/elasticsearch/guide/current/_wildcard_and_regexp_queries.html

Lucene Syntax : https://lucene.apache.org/core/2_9_4/queryparsersyntax.html

Check if exists or does not exist

You can add this by using the field list in the Events table.

1. Enter the name of the field in the Events table search - e.g. "originatingPeak" here

Fields	IDCSearchModes	ElapsedServiceTime	IDCNumberOfPaxRetrieved	TransactionStatus	Peak
originating	183			OK	AF
OriginatingPeak	4			OK	JL
	57			OK	2W
	7			OK	RO
	6			OK	VR
	2			OK	TP
	5			OK	LG
	2			OK	MS
	4			OK	VR
	2			OK	KE

2. Click on the field
3. Use the magnifier to add an "exists" filter to the query at the top - this adds a filter to pull out all rows where the field is populated : e.g. here the top level magnifier is clicked for OriginatingPeak.

Value	Action	Count / 500
1. KL	Q	238
2. SK	Q	100
3. LH	Q	59
4. AF	Q	29
5. LX	Q	20
6. 19	Q	8
7. SN	Q	8
8. WF	Q	6
9. EAF	Q	5
10. DY	Q	4

Terms

Pid (100%), Backend (100%), meta.flags (100%), ElapsedTransactionTime (100%), EventType (100%), FromSap (100%), Host (100%), HostName (100%), _id (100%), _type (100%), More

field must field : Source query : "CML"

time must field : sendTimestamp from : now-1h to : now

querystring must query : Phase: PRD

exists must field : OriginatingPeak

4. Similarly, you can click on the no-entry symbol to add a filter for does not exist - i.e. pull out all rows where the field is not populated

Filter by Peak

There is no easy way to wildcard and filter by Peak number.

(i think its potentially resource intensive and so at the Operations end the mappings/templates are not setup to support wildcarding for all fields - the only one wildcards work on is the Error text)

If you need to filter by Peak its best to use the **Host** or **HostName** fields - its cumbersome, but the only way currently.

The HostName always has the *_PEAKn_CML in the name, so the pie chart can give you the names to filter on. Alternatively, use the [SI viewer](#) and pull out the nodes for a CML Peak, and then use the node names in your query.

e.g. for CML Peak 5 add (HostName:425_PK5_CML OR HostName:119_PK5_CML OR HostName:524_PK5_CML OR HostName:618_PK5_CML OR HostName:223_PK5_CML OR HostName:323_PK5_CML) to the query

i.e. your full query becomes :

Source:CML AND Backend:* AND (HostName:425_PK5_CML OR HostName:119_PK5_CML OR HostName:524_PK5_CML OR HostName:618_PK5_CML OR HostName:223_PK5_CML OR HostName:323_PK5_CML)

QUERY

Source:CML AND Backend:* AND (HostName:425_PK5_CML OR HostName:119_PK5_CML OR HostName:524_PK5_CML OR HostName:618_PK5_CML OR HostName:223_PK5_CML OR HostName:323_PK5_CML)

FILTERING

field must field : Source query : "CML"

time must field : sendTimestamp from : now-1h to : now

querystring must query : Phase: PRD

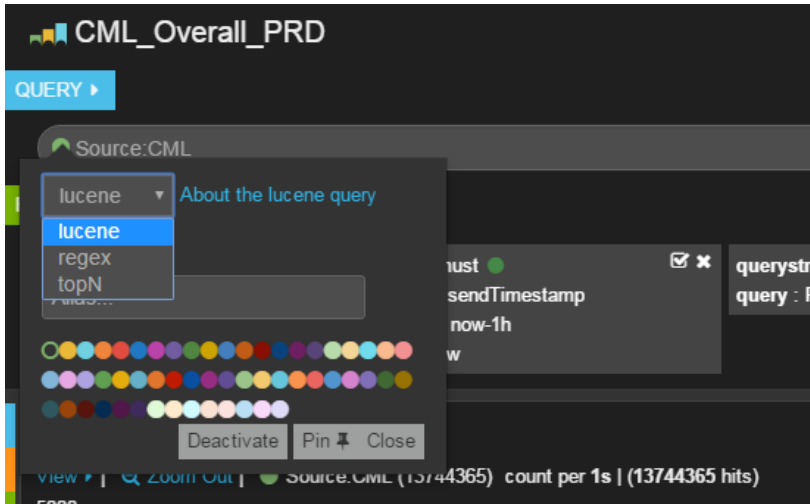
exists must field : OriginatingPeak

Choosing Query syntax as Lucene / REGEX / TopN

Click on the coloured dot in the query section and choose from a drop down

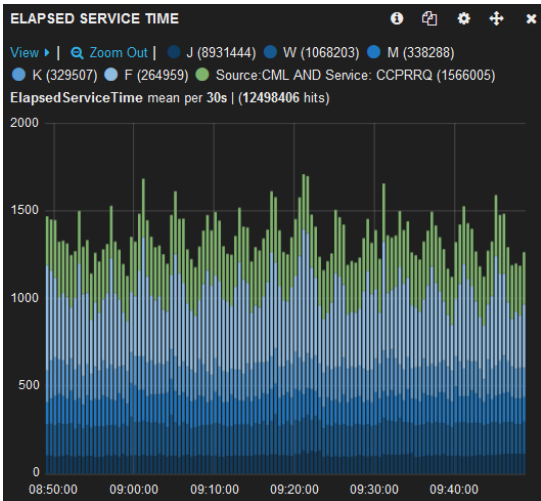
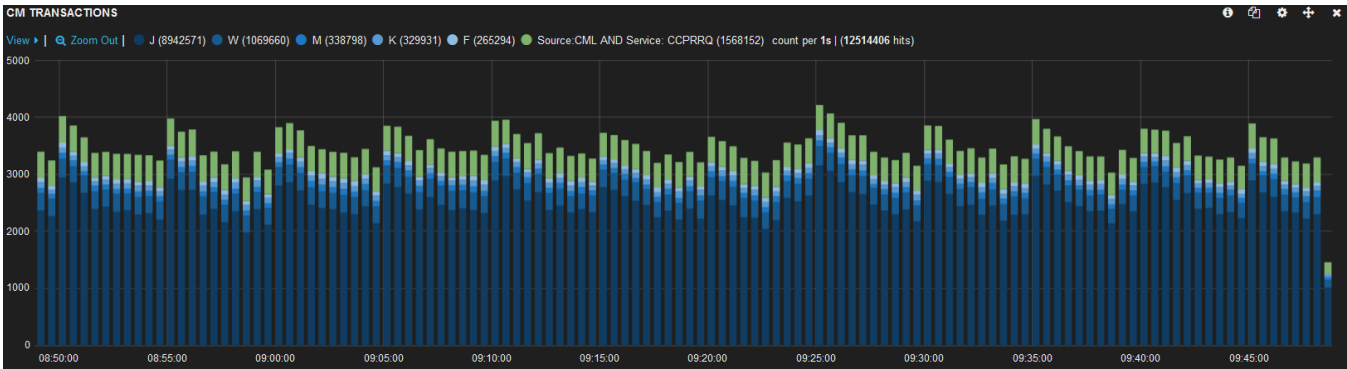
regex syntax links : <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-regexp-query.html>

Noted open issue on Kibana 3 - hardcodes the _all field for regex queries, leading to them not being usable : <https://github.com/elastic/kibana/issues/631>. Can do manual curl command though.

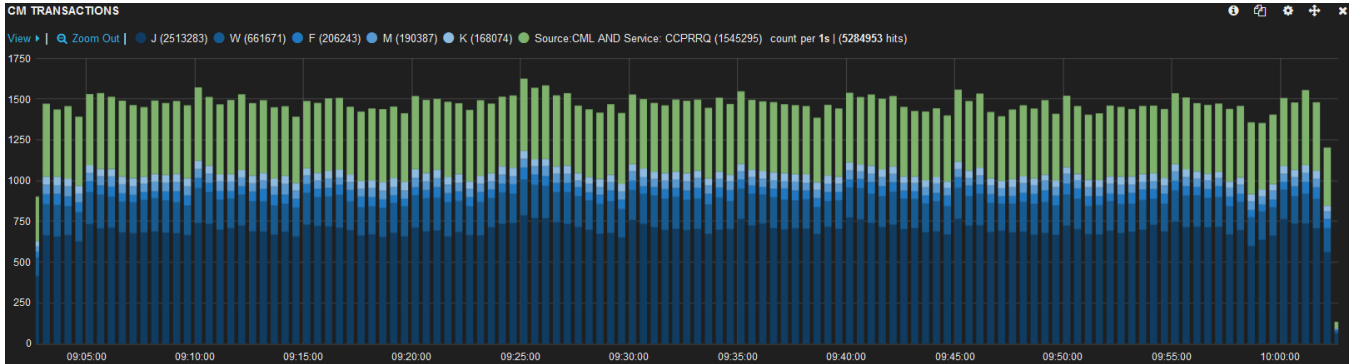
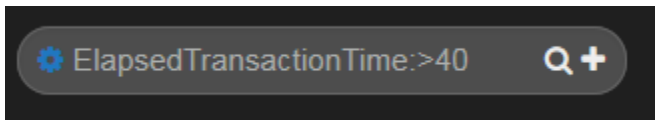


Using topN queries

The top N query finds the most popular terms in a field and uses them to compute new queries. In the example we would see the repartition of transactions per channel:

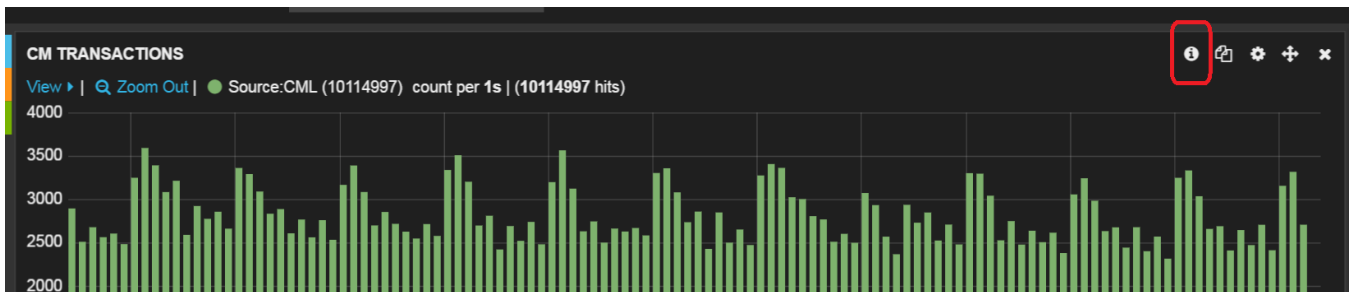
[illegible]

We can also add extra filters to the query that is built from that top N. For example, we could see what of those transactions have an ElapsedTransactionTime higher than 40ms:



Inspecting the query Kibana is building

Click the "i" on the Histogram



Fix Events Table not loading data issue

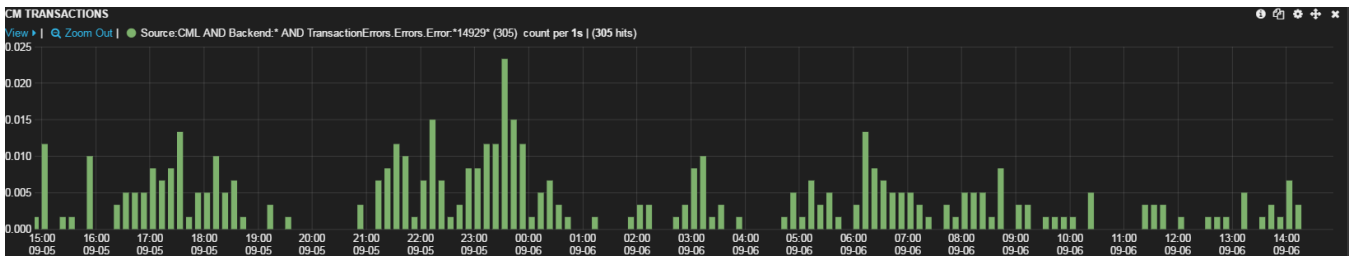
This is due to a specific ongoing bug (PTR 11736118)

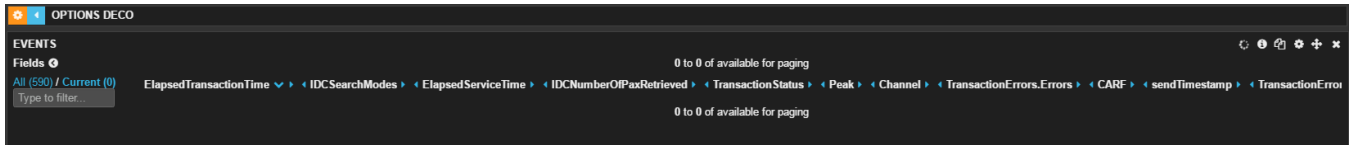
The Events table displayed in the bottom section of Kibana dashboards is useful to get information like the CARF, timestamp etc, which in turn helps you get the logs.

A field called ElapsedTransactionTime is not defined correctly as an integer on all nodes.

While Operations fix this, we notice there are some dashboards which erroneously try to sort the Event table by ElapsedTransactionTime - there is no good reason for doing this anyway.

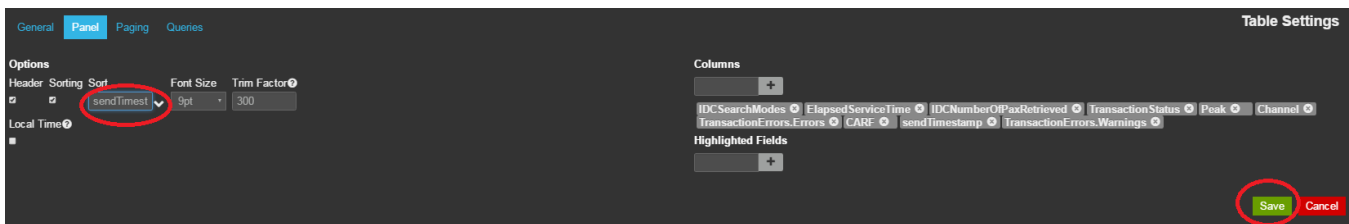
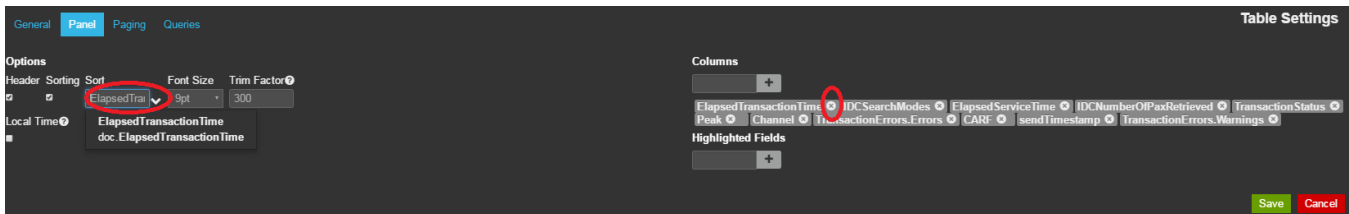
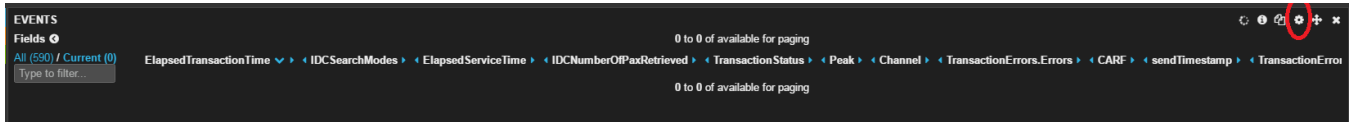
This results in exceptions, and no data is displayed in the Events table, though the Histogram at the top of the dashboard shows occurrences.





To fix this

- Click on the settings icon in the Events section
- Go to the Paging tab
- Remove the ElapsedTransactionTime from the Columns part, and change the Sort field to **sendTimestamp** instead of ElapsedTransactionTime
- Save, then Save the dashboard itself.
- Reload the dashboard



+ save the dashboard as usual (watch out for the green banner)

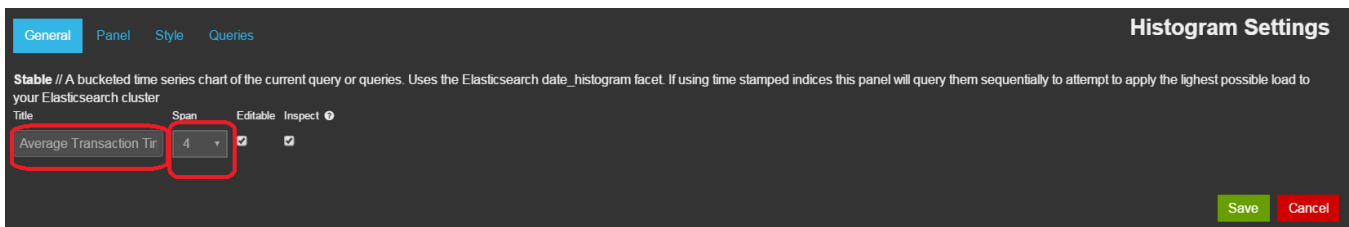
Charting the Elapsed time

Note any field you want to chart like this should be mapped as integer (refer TR 13185379: ACM:SWA3001:PRD: Use integers in functional monitoring as an example)

A good dashboard to use as an example is : http://umapv01.os.amadeus.net:8080/kibana-prd/#/dashboard/elasticsearch/CML_LoadMonitor_PRD

If your ElapsedTime field is mapped as an integer, this is how you can chart the Mean / Max times

1. Duplicate the main Transactions histogram so you get one more
2. Modify the properties of your new histogram - give it a Title and adjust the Span if you want on the General tab, Put a chart value of mean Or max on the Panel tab, put the value field on the Panel tab
3. Save, and then save your dashboard to get the green banner as usual



General
Panel
Style
Queries

Values

Chart value
Value Field

mean
ElapsedTransactionTime

Transform Series

Scale
Seconds
Derivative
Zero fill

1
0
0
1

Time Options

Time Field
Time correction
Auto-interval
Resolution

sendTimestamp
utc
1
100

Save
Cancel

If you want a chart of **max** values, just choose max above in the Chart value part.

A simple export

Exporting data is meant to be easy with future versions of Kibana, but for Kibana 3 we have a hack that may help.

Go to the bottom, to the events table and click on the "i" icon.

Copy the query , run from a linux box and just pipe the output to a .txt.

EVENTS

0 to 100 of 500 available for paging

Fields

Type to filter...

id

index

type

Backend

CARF

Channel

CMDiscrepancy

CMDiscrepancyInfo

sendTimestamp	Peak	FlightNumber	CMDiscrepancy	CARF	Prefix	NGIDiscrepancyInfo	NGIDiscrepancy	Trans
2017-03-15T12:59:56Z	AF	945	("AccNoSeat": "0", "DupeSeat": ...)	0001005Q72PLUG	005Q72PLUG0001:V8DPG94V6JWV...	("Data": {"H/A": UPI:2002D917...	("CMDiffNGI": "0", "CMSeatOnl...	OK
2017-03-15T12:59:50Z	AF	6134	("AccNoSeat": "0", "DupeSeat": ...)	0001005Q72PLSA	005Q72PLSA0001:E1Q824YR#RM...	("Data": {"ORY": UPI:2002A925...	("CMDiffNGI": "0", "CMSeatOnl...	OK
2017-03-15T13:00:45Z	AF	6104	("AccNoSeat": "2", "DupeSeat": ...)	0001005Q72PMCM	005Q72PMCM0001:WLLVVLVB2JV...	("Data": {"ORY": UPI:20030926...	("CMDiffNGI": "0", "CMSeatOnl...	OK
2017-03-15T12:59:57Z	AF	1567	("AccNoSeat": "0", "DupeSeat": ...)	0001001BR5VTH1	001BR5VTH10001:G6PKKBUD\$BW8...	("Data": {"FLR": UPI:2000B925...	("CMDiffNGI": "0", "CMSeatOnl...	OK

Last Elasticsearch Query

```
curl -XGET 'http://fmap606.os.amadeus.net:9230/funmon15032017/_search?pretty' -d '{
  "query": {
    "filtered": {
      "query": {
        "bool": {
          "should": [
            {
              "query_string": {
                "query": "Source:CML"
              }
            }
          ]
        }
      },
      "filter": {
        "bool": {
          "must": [
            {
              "fqquery": {
                "query": {
                  "query_string": {
                    "query": "Source:(\\\"CML\\\")"
                  }
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

```
amuni@lon1nx55:~/RGR_categoriser> >....
      "query_string": {
        "query": "SSMSRR"
      },
      "_cache": true
    },
  ],
}
```

...ensure you put the whole query, and just pipe the output to .txt.

```
    "NGIDiscrepancy": {  
      "order": "desc",  
      "ignore_unmapped": true  
    }  
  }  
]  
' > wai_ming.txt
```

Note links on the web to export (nothing tried - no time)

<http://stackoverflow.com/questions/18892560/is-there-any-way-in-elasticsearch-to-get-results-as-csv-file-in-curl-api?noredirect=1&lq=1>

<https://github.com/mobz/elasticsearch-head>

Other useful links with some examples available at [Querying ElasticSearch](#)

For kibana 4 : <https://github.com/minewhat/es-csv-exporter>

And an old PR for Kibana 3 : <https://github.com/elastic/kibana/pull/1463>

A useful tool used to export ES queries to CSV can be found is [es2csv](#).

The script the way it's implemented has the drawback that tries to store all the hits, so be careful with the query.

Otherwise it's quite easy to change it locally to eventually consider filters.