Integration of IT-DB Monitoring tools into IT General Notification Infrastructure

August 2014

0

Author: Binathi Bingi

Supervisor: David Collados Polidura

CERN openlab Summer Student Report 2014



1

Project Specification

The goal of this openlab summer student project was to standardize the service notification and alarming system in the IT Database group. For this we need to integrate the IT General Notification Infrastructure (GNI) into some of our database services, like for instance, RACMon, Enterprise Manager, Syscontrol, RMAN, Database on Demand, and Storage Administrators' tools. The objective was to make the GNI service our only mechanism to generate notifications and alarms (SMS, email, SNOW tickets) and as unique interface to visualize notifications.

Abstract

The IT Database group has independent monitoring tools/data and is immersed in a process of consolidating its monitoring infrastructure. The aim of this document is to provide insight into the way we achieved integration of GNI into our database services.

Table of Contents

Pr	oject Specification2
Ab	ostract2
1	Introduction4
	1.1 Database Monitoring tools4
	1.2 GNI and SNOW4
	1.3 Kibana4
2	Consolidation of IT-DB monitoring5
	2.1 Input5
	2.2 DB_notifier Producer6
	2.3 Messaging Broker10
	2.4 Output11
3	Visualizations11
4	Conclusion14
5	Future Proposals15
6	Bibliography15

1 Introduction

The IT Monitoring team is working to provide monitoring solutions in different areas.

Notifications: To get notified about problems affecting our services or nodes.

Archive: To archive data for offline batch analysis or historical reference.

Dashboards: To visualize monitoring data based on real time analysis.

To deliver successful monitoring solutions it is important to continuously monitor the status of all resources (network equipment, physical machines, virtual machines, operating systems, application services, etc.), to efficiently process all collected data, to promptly deliver monitoring results (notifications, alarms, reports, etc.) to the appropriate target, and to have the capability of executing complex queries across distinct monitoring data sets.

The IT-Database (IT-DB) group is in a process of consolidating the entire monitoring infrastructure. As a part of it we created a single mechanism (DB_notifier Producer) to generate notifications and alarms. We adopted a single interface Kibana, to visualize the monitoring data. Our goal was to stop sending notifications in the form of emails as much as possible and automate the opening of SNOW (Service Now) tickets for certain critical events.

1.1 Database monitoring tools

In IT-DB there are independent monitoring tools like Oracle Enterprise Manager (OEM), Syscontrol, RACMon, Lightweight monitoring tools, monitoring data/scripts for middleware tools like Syscontrol, or for other services like Storage, Backups, Recoveries, etc. These multiple monitoring applications are deployed for monitoring the status of specific resources. These were independent tools based on different tool chains. Despite their heterogeneity, they all shared a similar architecture and faced the same limitations, leading to unnecessary duplication of effort and increased difficulties in sharing monitoring data.

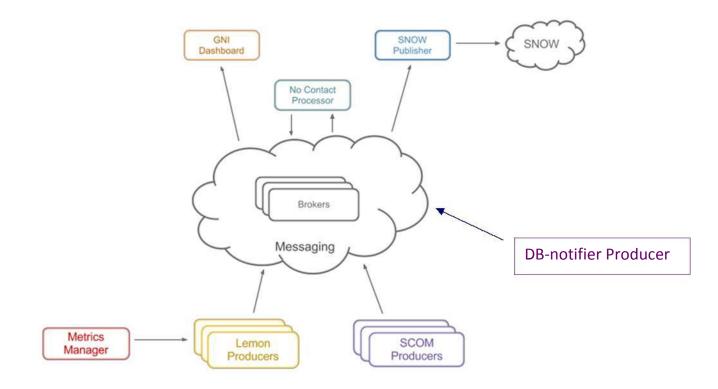
1.2 GNI and SNOW

GNI is a notification layer composed of several components responsible for dispatching alarms triggered in each data centre node to multiple operations tools. Different from other tools, GNI relies on dedicated transport layer, based on messaging brokers. Different producers publish notifications to the messaging infrastructure, which are taken by two consumers: one which creates tickets (SNOW tickets) in the CERN central ticketing system [1], while the second consumer populates a web application (Kibana) [2] showing current notifications.

1.3 Kibana

Kibana is a browser based analytics and search interface for Elasticsearch that was developed primarily to view Logstash event data. It is used by IT monitoring to visualize monitoring data in real time. Kibana is user friendly dashboard and can be configured to display data as per the user requirements. It facilitates query on the monitoring data and also to save the dashboards.

2. Consolidation of IT-DB monitoring



In the process of consolidation of monitoring tools, we created a DB-notifier producer which would take messages in a specific format as input from the monitoring tools to generate notifications. These notifications are sent to the messaging broker in the cloud. Further, the notifications are forwarded to the consumers like GNI and SNOW depending on the type of notification.

2.1 Input

DB_notifier Producer takes two arguments as input. Messages: In the form of key value pair. Configuration file: configuration files of monitoring tools generating the message.

Messages from Syscontrol monitoring tool are in the form of key value pair as shown below: Feb 12 16:15:37 itrac1111 SYSCONTROL_LOGGER[23286]: |V:002|BEGIN_REMOTE:17837630655748|USER_LOCAL:sysctl|CMD:ssh|USER_REMOTE:s ysctl|HOST:dbsrvg3301|SCRIPT:/etc/init.d/syscontrol|SINGLE:single|SUSPEND:|RUN_AS_RE MOTE:|IGNORE_ACTIVE:|TAG:tagenable_printers|DELAY:|DELAY_VAL:|SILENT:very_sil ent|HOST:dbsrvg3301|ENTITY:enable_printers|OPS:start|ARGS:|APPLY_PROFILE:|

DB_notifier Producer facilitates an option to provide either individual message or a log file of messages as input. Monitoring tools like OEM, RACMon generate messages in text format. We wrote a shell script that would consume these text messages and generate the message in required key value pair format. Whenever there is an event triggered in the monitoring tool, it would make a call to the shell script. The script would consume the text message, generate output in proper format and pass it as input to the DB_notifier producer.

The second argument passed to DB_notifier is the configuration file of the monitoring tool. We had written configuration files for each monitoring tool in IT-DB, these are passed as argument to the DB_notifier producer.

Example: Configuration file of the Syscontrol monitoring tool

[configuration] destination_dir = tmp/teststomp

. .

[header]		
Version	=	2.0
Туре	=	notification
des	=	/topic/monitoring.notification.db
env	=	dbnotifier_dev
host_group	=	physicsdb/pdbbackup
producer	=	dbnotifier_syscontrol
snow	=	1

2.2 DB_notifier Producer

DB_notifier producer is a python script that takes the configuration file of the monitoring tool and output of the shell script as input to generate a JSON(JavaScript Object Notation) file of the notifications. The notification specification [3] given by IT Monitoring must be followed in order for the GNI to accept the notification.

Message structure:

```
{
    "header":{ ... },
    "body":"{
        "payload": { ... },
        "metadata": { ... }
    }"
}
```

Notification specification mandates the inclusion of certain fields in the header and body of the messages as specified below. Current version of notification specification is v2.0

Header tags:

TAG	MANDATORY	ТҮРЕ	DESCRIPTION
m_version	yes	string	version of message specification
m_type	yes	string	type of message
			identifier of the application producing
m_producer	yes	string	messages
m_submitter_environment	yes	string	environment of the submitter host
m_submitter_hostgroup	yes	string	cluster of the submitter host
m_submitter_host	yes	string	submitter host
m_toplevel_hostgroup	no	string	top level hostgroup of the source host
m_snow	no	string	enable or disable snow tickets
destination	no		destination of the message (necessary for the stompclt)

Body/Metadata tags:

TAG	MANDATORY	ТҮРЕ	DESCRIPTION
timestamp	VAS	int	time when the event occurred (seconds
	yes	IIIt	since epoch)
uuid	yes	string	unique message identifier
metric_id	yes	int	metric id
metric_name	yes	string	metric name
entity	yes	string	source host
			hostgroup of the source host (snow
hostgroup	no	string	defaults to NO_HOSTGROUP)
environment	no	string -	environment of the source host
			is_essential flag of the source host (snow
is_essential	no	string	defaults to "0")
asset_id	no	string	identifier of the source machine
		. ,	detailed description of the notification,
			used in the SNOW title of the INC
description	no	string	(defaults to "no description provided" in
			snow)
notification_type	no	string	type of the notification (os, app, hw, nc)

state	no	string	local state of the notification: open, close, *
validity	no	int	notification validity in hours (default 24h)
egroup_name	no	string	egroup responsible for entity
fe_name	no	string	name of functional element to send snow ticket
			url or comment for troubleshooting
troubleshooting	no	string	information
snow_assignment_level	no	int	assigment level to created Snow ticket, defaults to 2
snow_grouping	no	string	boolean to request or not grouping of ticket in snow based on hostgroup and metric_name over a 1hour period, defaults to "1"
snow_instance	no	string	the CERN snow instance: cern, cerndev, cerntest, cernsandox
snow display value	no	string	the incident number of the record in
snow_display_value snow_id	no no		snow sys_id of already created snow ticket
			(event record)

Body/Payload tags:

TAG	MANDATORY	ТҮРЕ	DESCRIPTION
*	no	*	*

General format for a custom notification producer:

#/usr/bin/python

from monitoringdatamodel.gni import Notification

from monitoringdatamodel import utils

from messaging.message import Message

from messaging.queue.dqs import DQS

import time

import socket

if _____name___ == "____main___":

Local path to send message so stomp client can consume it

```
mq = DQS(path="/tmp/test")
# You can see the complete specification of a notification in:
my_notification_header = {
  'm_type': 'notification',
  'm_version': '2.0',
  'm_producer': 'mycustomscript',
  'm_submitter_environment': 'qa',
  'm_submitter_hostgroup': 'aimon/my/host/group',
  'm_submitter_host': socket.gethostname(),
  'destination': '/topic/monitoring.notification.generic'
}
my_payload = {
  "MyValues": [1, 2, 3],
  "custom": "field"
}
my_notification_body = {
  'payload': my_payload,
  'metadata': {
     'metric_id': 123456,
     'metric_name': 'my_big_problem_alarm',
     'entity': 'my_little_machine',
     'timestamp': int(time.time()),
     'destination': '/topic/monitoring.lemon.notification',
     'uuid': utils.generate_uuid(),
  }
}
notif = Notification({
  'header': my_notification_header,
  'body': my_notification_body
})
# We send the message to the queue directory
mq.add_message(notif.to_message())
```

The above producer has certain dependencies:

Monitoring-data-model: to build valid notifications[4].

python-messaging: for writing in the notifications queue of the broker[5].

stompclt: necessary dependency to send message to the messaging broker[6].

These dependencies are Red Hat Linux6 dependent rpms and can be installed using yum.

yes | yum install stompclt monitoring-data-model python-messaging

Now, we have a single notification producer for all the monitoring tools/data in IT-DB. The output of the notifier is sent to the messaging brokers using client that uses STOMP protocol.

2.3 Messaging Broker

stompclt is a versatile tool to interact with messaging brokers speaking STOMP and/or message queues on disk. It receives messages from an incoming module, optionally massaging them (i.e. filtering and/or modifying), and sends them to an outgoing module. Depending on which modules are used, the tool can perform different operations.

Here are the supported incoming modules:

- broker: connect to a messaging broker using STOMP, subscribe to one or more destinations and receive the messages sent by the broker
- queue: read messages from a message queue on disk

Here are the supported outgoing modules:

- broker: connect to a messaging broker using STOMP and send the messages
- queue: store the messages in a message queue on disk

In our case, stompclt is the messaging broker that forwards the JSON files from the DB_notifier Producer to the GNI. stompclt must be configured to the incoming queue path and is done in the stompclt-notifications.conf file.

Certain credentials are needed to access stompclt. They are configured in the stompclt-auth.conf file.

scheme = plain name = ask messaging team for one pass = ask messaging team for one

2.4 Output

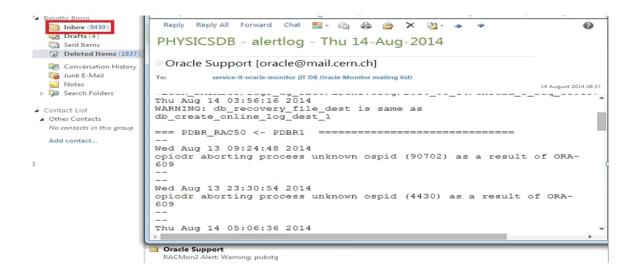
JSON format files are forwarded from the messaging brokers to the GNI Dashboards by a client.

Now, we can visualize the notification data in real time and perform query on it. In addition, generation of automatic SNOW tickets is facilitated for critical events.

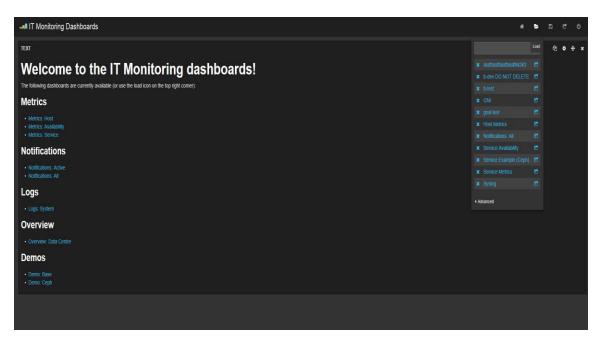
3. Visualizations

Kibana interface is used for visualization of data. We can login into dashboard to see, create and query on the monitoring data generated by IT-DB production monitoring tools.

The previous notification system was in the form of emails. In November 2013, we received almost 6638 emails, approximately 221 emails per day. It was very difficult to share or query the monitoring data using the previous notification system.



DB_notifier Producer has facilitated us to make use of the dashboards to visualize notifications rather than receiving them in form of emails.



Kibana interface

Many dashboards are available on Kibana. GNI dashboard can be loaded onto the Kibana.

🛤 GNI							6 ho	ours ago to a fe	w seconds ago 🕶	0	* !	b E	8	٥	Ċ
ERY • FILTERING • time must @ field : @timestamp from : now-Sh to : now	fiel	d <u>mustNot el</u> C C d : body.metadata.state ny : active ³	•												
HISTOGRAM															
HISTOGRAM	and Free Linder Lines												0 2	•	÷
view • 1 • (615) count p 40	per 5m ((615 NIS)														
30															
20						10.0									
10			A												
0 06:50 07:00 0	7:10 07:20 07:	30 07:40 07:50 08:00 0	18:10 08:20 08:30 08:40 0	08:50 09:00 09:10	09:20 09:30 09:4			11:00 11:10	11:20 11:30 11:4				12.2.9	12:30	12-40
08-19 08-19 0	8-19 08-19 08-	19 08-19 08-19 08-19 0	18-19 08-19 08-19 08-19 0	08-19 08-19 08-19	08-19 08-19 08-1	19 08-19 08-19 08-19 08-19 08-19	9 08-19 08-19	08-19 08-19	08-19 08-19 08-1	19 08-1	9 08-19	08-19	08-19 0	08-19	14.10
	8-19 08-19 08-	19 08-19 08-19 08-19 0 ● 02 \$ + ×	18-19 08-19 08-19 08-19 0		08-19 08-19 08-1 0 2	9 08-19 08-19 08-19 08-19 08-19	9 08-19 08-19 0 Ø		08-19 08-19 08- TOP METRICS	19 08-1	9 08-19	08-19	08-19 0 0 2		08-19
TOP PRODUCERS	8-19 08-19 08-									19 08-1	9 08-19	08-19			08-19 + ;
TOP PRODUCERS		● @ ◆ ÷ ×	TOP HOSTGROUPS		0 2 0 + ×	TOP ENTITIES	0 0	• + ×	TOP METRICS			08-19	0 2	•	08-19
TOP PRODUCERS Term lemon	Count	❶ ৫ ় ় ় ↔ × Action	TOP HOSTGROUPS Term	Count	0 2 0 + × Action	TOP ENTITIES Term	0 🕑 Count	O + × Action	TOP METRICS Term			08-19	0 @ Count	b O Act	08-19 + : ion Ø
TOP PRODUCERS Term lemon spectrum	Count 455	● ④ ◆ ÷ × Action Q ⊘	TOP HOSTGROUPS Term castor	Count 88	0 2 0 ÷ × Action Q Ø	TOP ENTITIES Term d3125-2v-ip57-shp5I-1614	0 2 Count 128	• + × Action Q 0	TOP METRICS Term MANAGEMENT_A	IGENT_L	OST	08-19	0 @ Count 138	Act Q	08-19 + : ion 0 0
TOP PRODUCERS Term lemon spectrum scom	Count 455 138		TOP HOSTGROUPS Term castor eos	Count 88 65		TOP ENTITIES Term d3125-2v-ip57-shp51-1614 br/sra24e03	0 2 Count 128 7	• + × Action Q 0 Q 0	TOP METRICS Term MANAGEMENT_A no_contact	AGENT_L4	OST	08-19	0 2 Count 138 89	Act Q Q	08-19 + , ion 0 0 0
TOP PRODUCERS Term lemon spectrum	Count 455 138		TOP HOSTGROUPS Term castor eos bi	Count 88 65 64	0 2 0 + x Action Q 0 Q 0 Q 0	TOP ENTITIES Term d3125-2x-ip57-shp5I-1614 bdrsa2403 teststack01-controller-cell01-01	• 2 Count 128 7 6	Image: Action X Image: Action	TOP METRICS Term MANAGEMENT_A no_contact exception_adaptec	\GENT_L4	OST	08-19	Count 138 89 58	Act Q Q Q	08-19 + , ion 0 0 0
TOP PRODUCERS Term lemon spectrum scom	Count 455 138		TOP HOSTGROUPS Term castor eos bl cloud_infrastructure	Count 88 65 64	0 2 0 + x Action Q 0 Q 0 Q 0	TOP ENTITIES Term d3125.2v1p57.shp51.1614 br/sra24a03 teststack01.controller.cal01.01 br/srg21a03.com.ch	• 2 Count 128 7 6	Image: Action X Image: Action	TOP METRICS Term MANAGEMENT_A no_contact exception_adaptec exception_c2_xrd	\GENT_L4	OST	08-19	0 2 Count 138 89 58 24	Act Q Q Q	08-19 + 1 ion 0 0 0
TOP PRODUCERS Term lemon spectrum soom Showing 3 out of 3 values	Count 455 138		TOP HOSTGROUPS Term castor eos bl cloud_infrastructure	Count 88 65 64	(2) (TOP ENTITIES Term d3125.2v1p57.shp51.1614 br/sra24a03 teststack01.controller.cal01.01 br/srg21a03.com.ch	• 2 Count 128 7 6	Image: Action X Image: Action	TOP METRICS Term MANAGEMENT_A no_contact exception_adaptec exception_c2_xrd	\GENT_L4	OST	08-19	0 2 Count 138 89 58 24	Acti Q Q Q	08-19 + ion 0 0 0 0
TOP PRODUCERS Term lemon spectrum scom Stowing 3 out of 3 values TABLE o	Count 455 138 22	0 2 0 + x Action Q 0 Q 0 Q 0	TOP HOSTGROUPS Term castor eos bl cloud_infrastructure	Count 88 65 64 61	Control Contro Control Control Control Control Control Control Control Control Co	TOP ENTITIES Term d3125 2+\p57-thp51-1614 bhrn24#03 testack01-controller-cal01-01 bhrn21#03 cem ch Showing 4 but of 156 values	0 20 Count 128 7 6 4	0 + x Action Q 0 Q 0 Q 0 Q 0	TOP METRICS Term MANAGEMENT_A no_contact exception adaptec exception c2_xrd_ Showing 4 out of 33 v	AGENT_L4	ost or		Count 138 89 58 24 0 2		+ : ion 0 0 0 0
TOP PRODUCERS Term Iemon spectrum soom TABLE @ Gitmestamp v > · · · ·	Count 455 138 22	0 2 0 + x Action Q 0 Q 0 Q 0	TOP HOSTGROUPS Term castor eos bi cloud_infrastructure Showing 4 out of 25 values	Count 88 65 64 61	(2) (TOP ENTITIES Term d31252-rip57-thp51-1614 bbm224tg3 feststack01-controller-call01-01 bfm214tg31 eststack01-controller-call01-01 bfm214tg31 est of 156 values 200 available for paging	Count 128 7 6 4 ion_type + < body.m	0 + x Action	TOP METRICS Term MANAGEMENT_A no_contact exception adaptec exception c2_xrd_ Showing 4 out of 33 v	AGENT_L4	ost or		Count 138 89 58 24 0 2 4 4		+) ion 0 0 0 0

GNI Dashboard

Fields O	0		200 1- 40	O of 555 available for paging		1.00		
All (168) / Current (55)				v or trop available for paging				
Type to filter	donotifier_syscontrol	level_arch_needisk-liert.,	2014-08-12716-16-20.000+02:00	Amp/TMP_INIT_man_backup	nnan_backup			orsch
8	donobilier_syscontrol		2014-08-12116-16-20.000+02:00	.tnp/TMP_INIT_ping_entit	ping_entities_cache			sysce
CD_M	dbnotfler_syscontrol	reto/ext.drayscontrol	2014-08-12716-16:20.000+02:00	.dnp/IMP_NIT_db-manager	db-manager-cron-sysc8			8990
D_mdex D_type	donotifier_syscontrol	level_arch_newdisk-lacc	2014-06-12716-16-20.000+02:00	.http://MP_INIT_man_backup	man_backup	14559150750324		orsc
Dody metadata description	stanotitier_syscentral	level_arch0_newdisk-i ca	2014-08-12716-16-20.000-12:00	top/TMP_NIT_man_backup	man_backup	820613766048		orac
D body metadata entity	donotifier_sysconbol	level_arch0_newdisk-in	2016-08-12716-16-20.000+02:00	.mp/TMP_WiT_man_backup	man_backup	11864419290750		orac
C body metadata metric_jid Dody metadata metric_name	dbrother_syscontrol		2014-08-12716-16:20.000+02:00	auti	enable_printers		dbsrvd260	
D body metadata Smeetamp	dbnobfler_syscontrol	service AISDEV -RR -p -L.	2014-08-12716-16-20.000+02:00	8671	ohs_ais06_dev		dovrtg8008	
D body metadata usid: D body payload	donobilier_syscontrol	-service AISDEV -RR -p -I	2014-06-12716-16:20.000+02:00	sah	ohs_ais08_dev		dbvrtg4208	
D body payload APPLY_PROFILE	donabilier_syscontrol	service AISDEV /RR -p (L)	2014-05-12116-16-20.000+02-00	esh.	otis_als00_dev		dbvrlg4007	
ST : Dody payload BEOM LOCAL	donotifier_syscontrol		2014-06-12716-16-20.000+02:00	sah	enable_printers		dbsrvd203	
D body payload BEGIN_REMOTE	donotitier_syscontrol		2014-06-12716-16:20.000+02:00	esh.	adhoc_mon		dbervg3410	
D body payload BKG	donobiliar_syscontrol	ieto/mit.disyscontrol -s	2014-08-12716-16:20.000+02:00	http:/TMP_ENIT_db-manager	db-manager-cron-oracle	17231293741200		. 0190
Dody payload CMD_SOURCE	donother_syscontrol		2014-08-12T18-16:20.000+02:00	esh	adhod_mon		dbervg3410	
C Dody payload DELAY	donotifier_syscontrol	-service AlSPROD -RR -p	2014-08-12716-16-20.000+02:00	ash.	ofis_ais08_prod		ftrac1207	
Dody payload DELAY_VAL	donother_syscentral		2014-06-12716-16:20.000+02:00	960	athor_mon		dbervg3301	
D body payload EXIT	dbnotiter_syscontrol		2014-08-12716-16-20.000+02:00	waft-	edmaGronsProd		dbsrvd255	
D body payload FINGH REMOTE	dbrotiter_syscontrol	reto/wit drayscontrol -+	2014-06-12718-16-20.000+02:00	.tmp/TMP_INIT_db-manager	dD-manager-cron-sysc8			syst
8	donother_syscontrol	level_arch_newdisk -i zor	2014-58-12716-16:20.000+02:00	.tmp/TMP_INIT_man_backup	man_backup			orac
C body payload X0NORE_ACTIVE	donother_syscentral	level_arch_headlak-Ledh	2014-06-12716-16:20.000+02:00	http://www.tvitt_man_backup	man_backup	8420576957265		orac
D body payload OPS	donotiter_syscontrol	Atc/init.dityscontrol -s	2014-06-12718-18-20.000+02:00	/mp/TMP_INIT_db-manager	db-manager-cron-oracle	1472221174512		orse
D body payload RUN_ALL REMOT	donotifier_syscontrol		2014-00-12716-16:20.000+02:00	ash	eamsCronsProd		0berv0255	
Donty payload SCRIPT	donother syscontrol	level and needlok i acc	2014-05-12716-16-20.000+02:00	Anp/TMP INIT man backup.	man backup	753664891920		one

Dashboards also display various fields regarding each event. It is possible to choose the fields to be displayed for each event and query using the value of the fields.

	reference	/_plugin/kibana/#/	dashboar	d/elastic	search/GNI		🗇 🗸 🕵 🚺 🚺 🗸 Google		#9	2
ERING •	œ× fiel	d must a	17	x 0						
ield : @timestamp rom : now-7d p : now	fiel	d :hody.metadata.sta iry :"open"								
OSTGROUPS 🖄 🌢 🕯	÷ ×	METRICS	ବ ୦	+ ×	:	N				
erm Count Action	,	Term	Coun	Action		Ø				
owing 0 out of 0 values		Syscontrol_logger_ cal	lo 1	90						
		Showing 1 out of 1 va	lues							
TA								0 th	• +	
								0 2	• +	
lds O						0 to 1 of 1 available for paging		0 2	• +	
elds G 167) / Current (42)	2014-08	-07T14:18:13.000+02	:00	db	notifier_syscantrol	0 to 1 of 1 available for paging politibility cern.ch	Syscontrol_logger_local	o &	• +	
telds @ 167) / Current (42) ype to filter	View: T	-07T14:18:13.000+02 ible / JSON / Raw					Syscontrol_logger_local			
elds I (42) 167) / Current (42) ype to filter	View: T	ible / JSON / Raw	4	Action	Value		Syscontrol_logger_local			
Ids () 167) / Current (42) ype to filter Sinducano Jad	View: T	ible / JSON / Raw	4				Syscontrol_logger_local			
olds () 167) / Current (42) ype to filter (2) ministano Id Jonex Jypo	View: Tr Field @time: _id	ible / JSON / Raw	i I	Action	Value		Syscontrol_logger_local			
bids G 167) / Gurrent (42) ype to filter Jd 	View: Tr Field @time:	ible / JSON / Raw		Action Q ⊘ Ⅲ	Value 2014-08-07T12:18:13.000Z Zo4Je_MASd6JbwS-Ohe_1Q		Syscentrol_logger_local			
elds © (167) / Current (42) ype to filter _id _index _ype body metadata description (cody metadata metric, id	View: Tr Field @time: _id	ible / JSON / Raw		Action Q Ø Ⅲ Q Ø Ⅲ	Value 2014-08-07T12:18:13.000Z Zo4Je_MASd6JbwS-Ohe_1Q		Syscontrol_logger_local			
ATA leids O (167)/Current(42) Type to filter 2 d 3 d 3 d 3 d 3 d 3 d 3 d 4 d 5 d 4 d 5 d 4 d 5 d 4 d 5 d 5 d 5 d 6 d 6 d 6 d 6 d 6 d 6 d 6 d 6	View: Tr Field @time: _id _index _type	ible / JSON / Raw	,	Action Q 0 III Q 0 III Q 0 III	Value 2014-08-07T12:16:13.000Z Zo4Je_MASd6JbwS-Ohe_1Q gnl		Syscontrol_logger_local			

We can also query for top host groups or producers, etc. as per the requirement and configure how the result of our query would be displayed (either in the form of tables, or graphs like histograms, pie charts, etc.).

							12 hours ago	to 3 minutes	ago 👻	c •	1	5	et •	• ¢
dbnotifier_syscon	trol													Q.
LTERING >														
time must field : @timestamp from : now-12h to : now	ž	X field mi field : b- query : i	ody.metadati		*									
GRAPHS														
TOP PRODUCERS	Count	Action	TOP H	OSTGROUPS	0 0	Action	TOP ENTITIES	Count	Action	t TOP I	METRICS	0	Coun	-t-
dbnotifier syscontrol	11126	90	005	127		90	pcitdb07.cem.ch	11126	90	Sysc	ontrol logge	r local	8183	a
lemon	586	90	castor			90	bxtsm518	43	0.0	1000	ontact		128	a
scom	6	90	backu	p 81		90	lxtsm519	27	90		tion.scsi_b	lockdevice	d 81	9.9
Showing 3 out of 3 values			Showin	64 g 4 out of 25 valu	0 5	9	Ibtestvobox Showing 4 out of 139	18 values	90	exce	error_report stion_adapte	c_bbu_erro	or 39	QG
TABLE Fields O		@timestam		• header.m_pro	ducer •	• body.metadata	0 to 100 of 500 available.	ble for paging	• body.me	adata.metric	_name •	€ →	🕙 💿	+ ronmen
All (225) / Current (55)		2014-08- 14T08:58:12	000+02:00	dbnotifier_sysco	ntrol	pcitdb07.cem.ch	pRxD-SN3RN2P	JVaXjFs.dxQ	Syscontrol	logger_local				
🖌 comentamo		2014-08- 14T08:58:12	000+02:00	dbnotifier_sysco	ntrol	pcitdb07.cem.ch	RnscLEtcS-irQM	8tHfr7ag	Syscontrol	logger_local				
M		2014-08- 14T08:58:12	.000+02:00	dbnotifier_sysco	ntrol	pcitdb07.cem.ch	GJYW6IBGRUWb	d0h4Fb6zlA	Syscontrol	logger_local				
 Anody mechanism dencer Decity mechanism and y 	ption	2014-08- 14T08:58:12	000+02:00	dbnotifier_sysco	ntrol	pcitdb07.cem.ch	TucXcCMnQ2a9	955Tb55Dpw	Syscontrol	logger_local				
 Incody most act at a modern Incody most act at a modern 		2014-08-		dbnotifier_sysco	ntrol	pcitdb07.cem.ch	U9ohql1WS6Cil	JU94F4Yz4A	Syscontrol	logger_local				

For example if we run a query for top producers on GNI dashboard and configure to display result in the form of a pie chart. Clicking on any of the sectors of the pie chart would display the concerned notification data.

4. Conclusion

Through this project we achieved

Single IT-DB producer: DB_notifier Producer that we developed is the single notification producer for all the IT monitoring tools in IT-Databases. It is easy to maintain and run same notifier in all the production monitoring machines.

Visualize and query monitoring data: Display of data on Kibana interface facilitates visualization of data in the form of graphs and tables. It is easy to query or filter the monitoring data using the tools available on Kibana.

Configure and display monitoring data: Dashboards can be configured to display the monitoring data that interests the user. These dashboards can be saved and then shared, imported or exported.

Email notifications: We could stop global email notifications to a considerable level by putting DB_notifier Producer into production. Events, alerts, etc. are displayed on the dashboards rather than sending them in the form of emails to the group users.

SNOW tickets: Automatic Service NOW tickets can be opened for events or alerts that are critical enough.

5. Future Proposals

At present, production monitoring machines (db-manager, oem.cern.ch, etc.) run on RedHat Linux5. DB_notifier Producer that we developed has RedHat Linux6 dependencies. Servers need to be migrated to RedHat Liux6 in order to bring DB_notifier Producer into production. We also need to identify the messages that should generate SNOW tickets and put it into production. Currently we integrated Syscontrol, OEM into GNI and are working on RACMon. We can further add more IT-DB monitoring tools into GNI.

6. Bibliography

- 1. <u>https://cern.service-now.com/service-portal/</u>
- 2. <u>https://dashboards.cern.ch/public/_plugin/kibana/#/dashboard/elasticsearch/Notificati_ons:%20All</u>
- 3. https://itmon.web.cern.ch/content/build-custom-notifications-consumer
- 4. http://itmon-doc.web.cern.ch/itmon-doc/monitoring-data-model/
- 5. https://messaging.readthedocs.org/en/latest/index.html
- 6. http://mig.web.cern.ch/mig/doc/stompclt.html