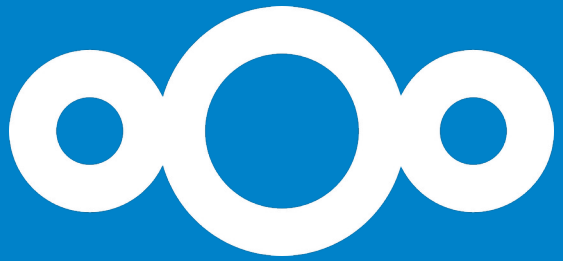


An overview of Nextcloud Monitoring

Pietro Marini

Senior Sales Engineer

08 March 2023

CS3 2023, Barcelona

Agenda

1. Introduction
2. Monitoring by layer
   - Application
   - Web Server
   - Database
   - Storage
3. Sample Metric Pipeline: User-agent
4. Conclusion and Q&A

Introduction

Introduction

Monitoring-by-layer approach: the goal is to have a separated monitoring dashboard per layer and host.

- Nextcloud Application on top of a LAMP stack
- Today scope: Nextcloud Files and observational time-series monitoring
- Assuming the recommended stack
  - Web Server: Apache2
  - DB: MariaDB
  - OS: Ubuntu Server 22.04
- Using Zabbix as IT infrastructure monitoring tool

Monitoring by Layer

Application

What to monitor?

- Total storage available in data directory
- Users: total, active
- Largest db tables in terms of records or size on disk
- Number of shares (by type)
- Requests by User-agent

Goal

- Follow the behaviour of metrics over time, particularly the rate of growth
- Understand how the users are using the application
- Find any deviation from the expected behaviour

How to extract metrics?

Use endpoint exposed by `serverinfo: ocs/v2.php/apps/serverinfo/api/v1/info`

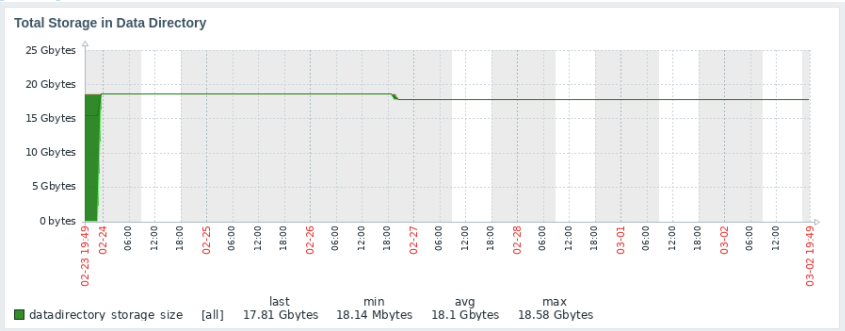
1) Create a token: `occ config:app:set serverinfo token --value mytoken`
2) Use an HTTP client to fetch metrics (json or xml output), ex:
   `curl -H 'NC-Token':'mytoken' https://my-nc.com/ocs/v2.php/apps/serverinfo/api/v1/info?format=json`
3) Collect and process

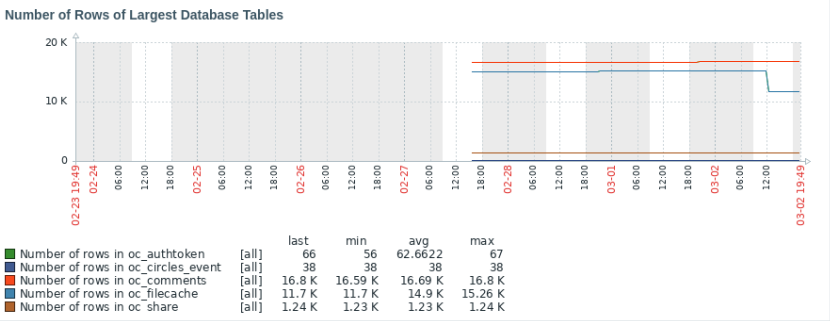
Other metrics available via Web Server log processing, database queries and audit log

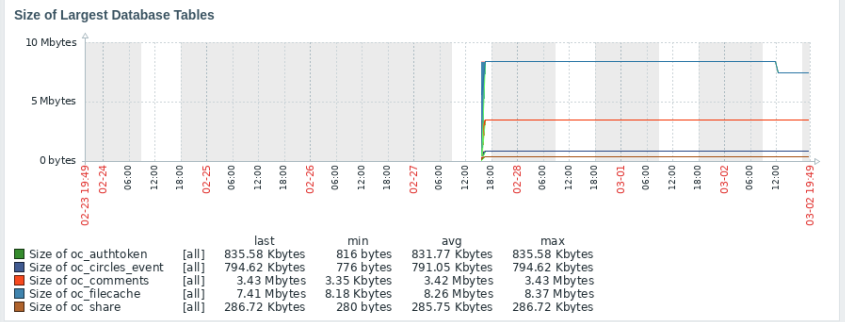
Application

Sample Nextcloud application monitoring dashboard showing some important application metrics, such as the number of files, the total storage in the data directory and number of shares per type. This dashboard has been built using Zabbix.

Application







Database

What to monitor?

- Number of queries (per type SELECT, INSERT/UPDATE, DELETE)
- Input/output traffic from/to Web Server(s)
- Slow queries
- InnoDB buffer pool memory usage
- Number of available connections

Goal

- Ensure optimal performance
- Ensure database engine keeps up with application usage

How to extract metrics?

Zabbix agent

1) Install and configure the Zabbix agent
2) Use template MySQL by Zabbix agent

Slow queries

1) Enable slow query logging (mysql console or config file)

```
slow_query_log          = 1
slow_query_log_file     = /var/log/mysql/mariadb-slow.log # can also write to db table
long_query_time         = 2 # seconds
log_slow_verbosity      = query_plan,explain
```

2) Analyze it using the utility `mysqldumpslow`

```
$ mysqldumpslow -t 3 -r /var/log/mysql/mariadb-slow.log
```

Database


Dashboard MySQL performance
All hosts / chat.x51-hserver.com / MySQL performance
Zoom out
Last 7 days
MySQL: Operations
MySQL: Command Delete per second [avg] last 0.01752 min 0 avg 0.02617 max 68.4205
MySQL: Command Insert per second [avg] 0.006394 0 0.005533 1.5449
MySQL: Command Select per second [avg] 2.4054 0.2335 2.2971 139.1716
MySQL: Command Update per second [avg] 0.5598 0 0.5958 3.8539
MySQL: Queries
MySQL: Queries per second [avg] last 3.555 min 0.4336 avg 3.4868 max 143.6492
MySQL: Questions per second [avg] 3.555 0.4336 3.4868 143.6492
MySQL: Slow queries per second [avg] 0 0 0.00001285 0.06696
Trigger: MySQL: Server has slow queries [> 3]
MySQL: Connections
MySQL: Aborted clients per second [avg] last 0 min 0 avg 0.000009165 max 0.01679
MySQL: Aborted connections per second [avg] 0 0 0.004805 0.9287
MySQL: Connections per second [avg] 0.1878 0.08079 0.1956 3.1143
MySQL: Max used connections [avg] 16 11 13.0464 16
Trigger: MySQL: Server has aborted connections [> 3]
MySQL: Bandwidth
MySQL: Bytes received [avg] last 645.7707 Bps min 62.825 Bps avg 600.3941 Bps max 37.5 KBps
MySQL: Bytes sent [avg] 14.19 KBps 1.17 KBps 9.59 KBps 904.04 KBps
MySQL: InnoDB buffer pool
MySQL: InnoDB buffer pool pages free [avg] last 4.8 K min 4.8 K avg 5.01 K max 5.3 K
MySQL: InnoDB buffer pool pages total [avg] 8.11 K 8.11 K 8.11 K 8.11 K
MySQL: InnoDB buffer pool read requests per second [avg] 297.0821 14.7597 217.6288 17.03 K
MySQL: InnoDB buffer pool reads per second [avg] 0 0 0.0001884 0.579
MySQL: Threads
MySQL: Threads cached [avg] last 13 min 1 avg 10.4605 max 15
MySQL: Threads connected [avg] 1 1 1.0987 9
MySQL: Threads created per second [avg] 0 0 0.00001649 0.08365
MySQL: Threads running [avg] 1 1 1 2
Zabbix 6.2.7. © 2001–2023, Zabbix SIA

Web Server

What to monitor?

- Number of requests per second
- Bandwidth
- Response Time
- Status of web server workers
- Number of workers

Goal

How to extract metrics?

1) Enable Apache2 module mod_status
2) Add the following section in the Virtual Host

```
<Location "/server-status">
   SetHandler server-status
   Require ip <Ip of the monitoring host>
</Location>
```

3) Add the following line to the Nextcloud .htaccess (last <IfModule> section)

```
RewriteCond %{REQUEST_URI} !=/server-status
```

4) Use Zabbix template Apache by HTTP


Apache performance
Dashboard
Apache performance
All hosts / chat.x51-hserver.com / Apache performance
Zoom out
Last 7 days
Apache: Requests per second
Apache: Requests per second [avg] last 0.1378 min 0.01657 avg 0.09728 max 2.9557
Apache: Workers total
Apache: Total workers idle [avg] last 8 min 3 avg 7.8729 max 10
Apache: Total workers busy [avg] last 1 min 1 avg 1.1864 max 14
Apache: Bytes per second
Apache: Bytes per second [all] last 3.91 KBps min 101.7851 Bps avg 38.27 KBps max 22.74 MBps
Apache: Worker states
Apache: Workers DNS lookup [avg] last 0 min 0 avg 0 max 0
Apache: Workers idle cleanup [avg] last 0 min 0 avg 0 max 0
Apache: Workers logging [avg] last 0 min 0 avg 0 max 1
Apache: Workers reading request [avg] last 0 min 0 avg 0 max 3
Apache: Workers sending reply [avg] last 1 min 1 avg 1.0085 max 11
Apache: Workers starting up [avg] last 0 min 0 avg 0 max 0
Apache: Workers closing connection [avg] last 0 min 0 avg 0 max 4
Apache: Workers finishing [avg] last 0 min 0 avg 0 max 0
Apache: Workers keepalive (read) [avg] last 0 min 0 avg 0.07627 max 8
Apache: Workers slot with no current process [avg] last 139 min 131 avg 139.8983 max 145
Apache: Workers waiting for connection [avg] last 8 min 3 avg 7.8729 max 10
Apache: Service response time
Apache: Service response time [all] last 53.8ms min 0 avg 53.51ms max 1s 65.85ms
Trigger: Apache: Service response time is too high [> 10]
Apache: Service ping
Apache: Service ping [all] last 1 min 0 avg 0.9939 max 1
Trigger: Apache: Service is down [= 0]

Apache Server Status for [redacted] (via 10.36.21.154)

Server Version: Apache/2.4.52 (Ubuntu) OpenSSL/3.0.2
Server MPM: prefork
Server Built: 2023-01-23T18:34:42

Current Time: Friday, 03-Mar-2023 17:20:07 CET
Restart Time: Thursday, 02-Mar-2023 13:09:06 CET
Parent Server Config. Generation: 2
Parent Server MPM Generation: 1
Server uptime: 1 day 4 hours 11 minutes
Server load: 0.81 0.59 0.61
Total accesses: 10626 - Total Traffic: 6.6 GB - Total Duration: 16367126
CPU Usage: u90.19 s53.72 cu688.46 cs162.14 - .98% CPU load
.105 requests/sec - 68.1 kB/second - 0.6 MB/request - 1540.29 ms/request
1 requests currently being processed, 9 idle workers

```
._W._.__._._.....................................................
.................................................................
.....................
```

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
0-1	-	0/0/958	.	0.00	1182	0	808770	0.0	0.00	7.73	::1	http/1.1	[redacted]:80	OPTIONS * HTTP/1.0
1-1	1049131	0/32/819	_	2.49	33	1	892817	0.0	0.21	10.48	178.60.198.221	http/1.1	[redacted]	
2-1	1043460	0/133/834	_	10.27	1	48	1158725	0.0	1.26	334.36	31.44.174.158	http/1.1	[redacted]:443	GET /ocs/v2.php/core/navigation/apps?absolute=true&format=json
3-1	1043592	0/130/811	_	19.40	34	1	3663895	0.0	1897.78	2207.71	178.60.198.221	http/1.1	[redacted]:443	GET /server-status?auto HTTP/1.1
4-1	1043510	0/191/1079	W	30.99	0	0	1684492	0.0	972.09	1232.48	178.60.198.221	http/1.1	[redacted]:443	GET /server-status HTTP/1.1
5-1	-	0/0/848	.	0.00	1201	0	1464327	0.0	0.00	768.27	::1	http/1.1	[redacted]:80	OPTIONS * HTTP/1.0
6-1	1049409	0/27/753	_	2.31	32	1	857932	0.0	0.17	5.75	178.60.198.221	http/1.1	[redacted]	
7-1	-	0/0/652	.	0.00	1184	0	713010	0.0	0.00	176.21	::1	http/1.1	[redacted]:80	OPTIONS * HTTP/1.0
8-1	1049410	0/41/571	_	2.69	17	109	892005	0.0	0.22	334.50	31.44.174.158	http/1.1	[redacted]:443	PROPFIND /remote.php/dav/files/[redacted]i/ HTTP/1.1
9-1	1049010	0/35/375	_	3.32	49	123	547811	0.0	0.26	7.02	10.36.21.154	http/1.1	[redacted]:443	GET /ocs/v2.php/apps/serverinfo_hetzner/api/v1/info?format=json
10-1	-	0/0/662	.	0.00	1204	0	706314	0.0	0.00	309.28	::1	http/1.1	[redacted]:80	OPTIONS * HTTP/1.0
11-1	1016961	0/403/592	_	58.02	51	123	907004	0.0	918.31	1222.10	10.36.21.154	http/1.1	[redacted]:443	GET /ocs/v2.php/apps/serverinfo_hetzner/api/v1/info?format=json
12-1	1049007	0/53/502	_	4.17	49	88	980877	0.0	3.04	35.24	31.44.174.158	http/1.1	[redacted]:443	PROPFIND /remote.php/dav/files/[redacted]/ HTTP/1.1
13-1	-	0/0/753	.	0.00	1203	0	700646	0.0	0.00	98.40	::1	http/1.1	[redacted]:80	OPTIONS * HTTP/1.0
14-1	1049434	0/34/354	_	2.77	31	52	242480	0.0	0.19	2.09	31.44.174.158	http/1.1	[redacted]:443	GET /ocs/v2.php/core/navigation/apps?absolute=true&format=json
15-1	-	0/0/45	.	0.00	1202	0	141709	0.0	0.00	0.24	::1	http/1.1	[redacted]:80	OPTIONS * HTTP/1.0
16-0	-	0/0/18	.	0.00	87308	0	4306	0.0	0.00	0.10	::1	http/1.1	[redacted]:80	OPTIONS * HTTP/1.0

Apache server status as exposed in the browser by the mod_status module. It shows several performance and status metric along with a legend that explains how to read them. Output is also available as JSON for scripting.

Storage

What to monitor?

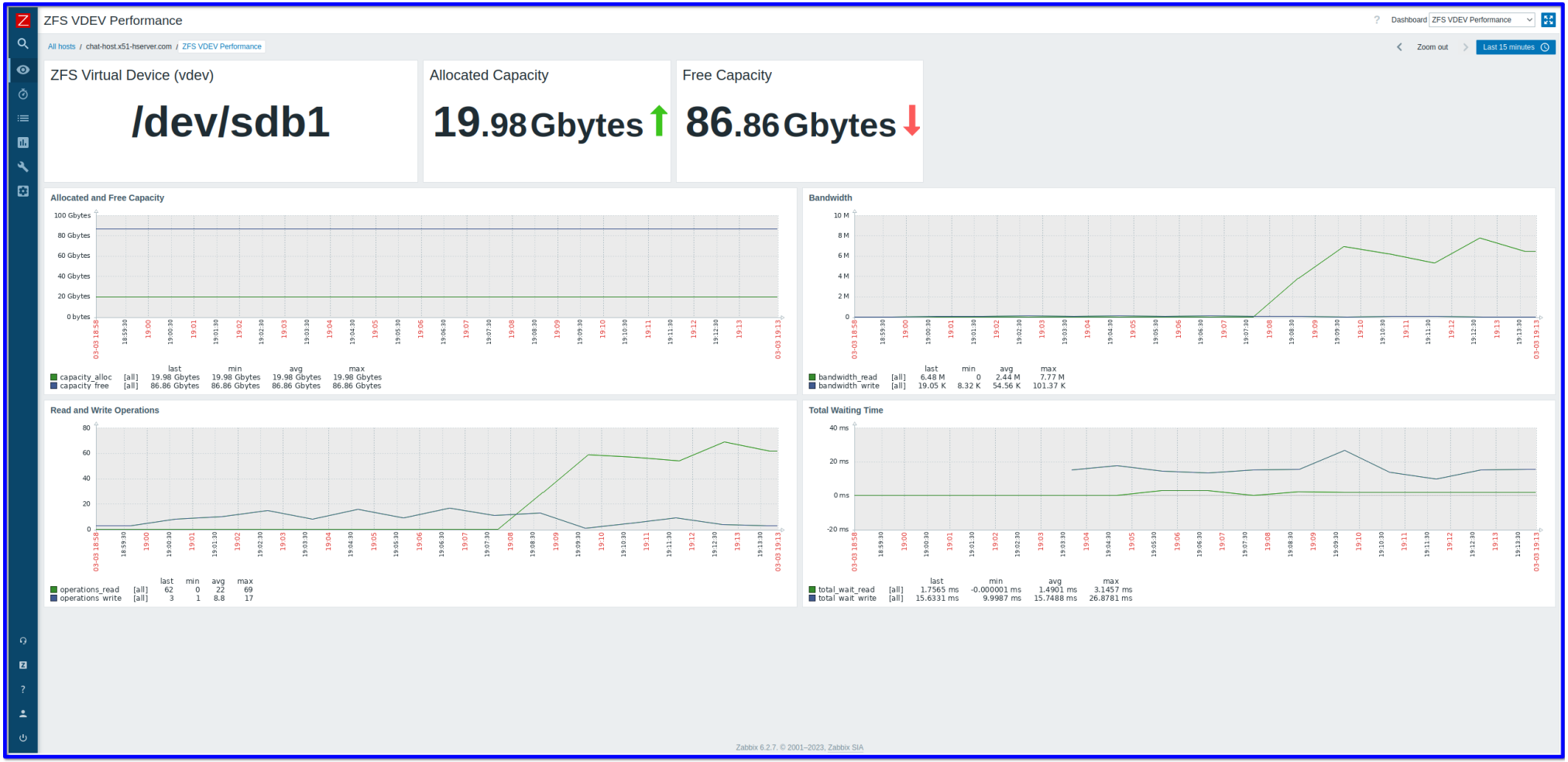
- IOPS
- Average waiting time per read/write operation
- Monitor at storage software layer (ZFS, LVM, Ceph..) and/or at block device level

How to extract metrics?

- On zfs, you can use zpool iostat

```
$ man zpool iostat
```

- Options allow you to format the output for batch / automated processing
- Process the output with Zabbix or with a scripting language
- Build and setup your own Zabbix template



Performance dashboard of the virtual device that hosts the data directory of a Nextcloud instance

System

What to monitor?

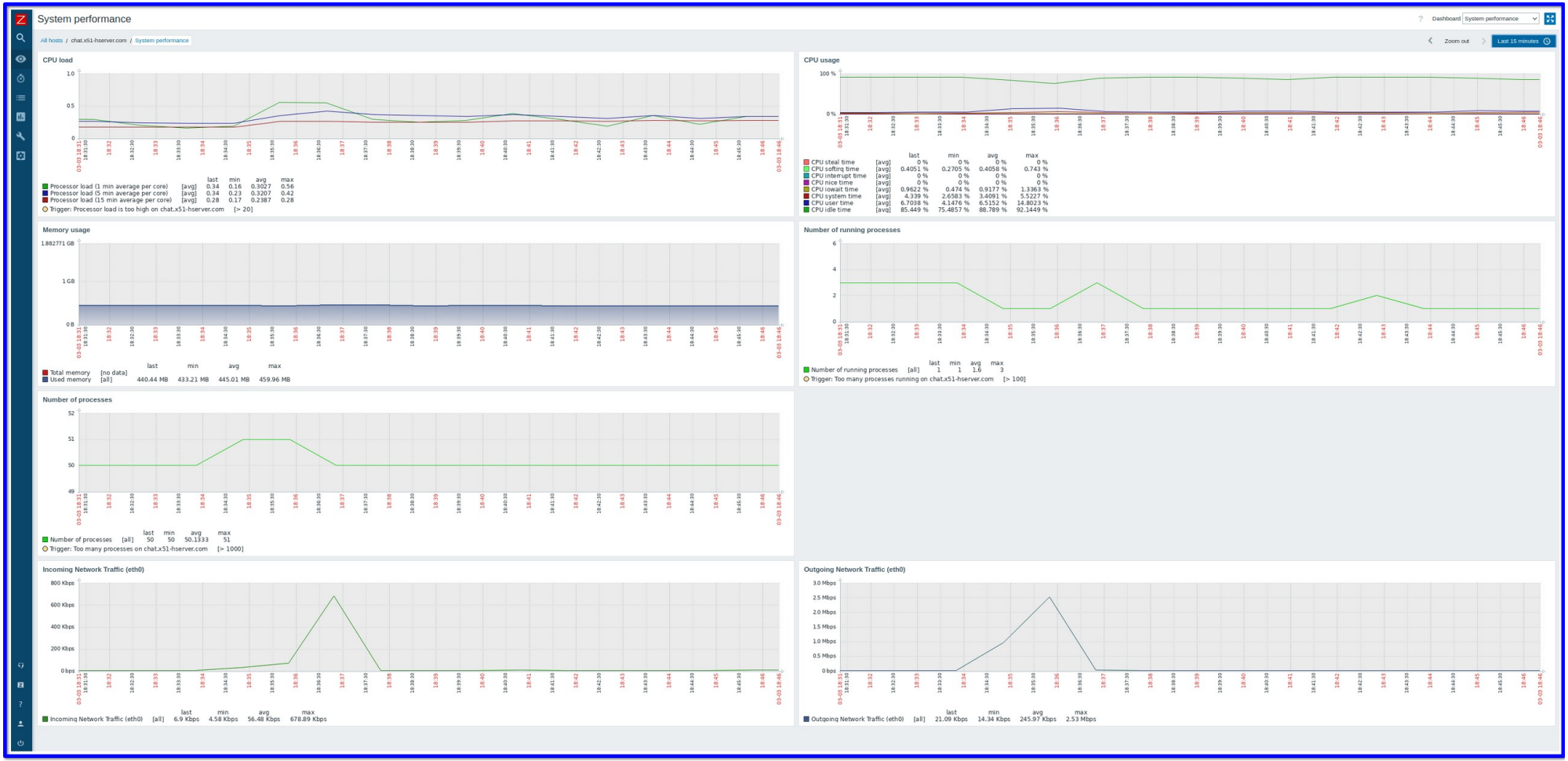
- Hardware resources: CPU, Memory, Swap, Storage, Network Bandwidth
- For all the core systems: Hypervisor (if any), Web server, Database, Cache, Storage system, Load Balancer

How to extract metrics?

1) Use Zabbix template Linux by Zabbix Agent

You can try to find a customized template for your Linux distribution or other supported OS. If it doesn’t exist, build one.

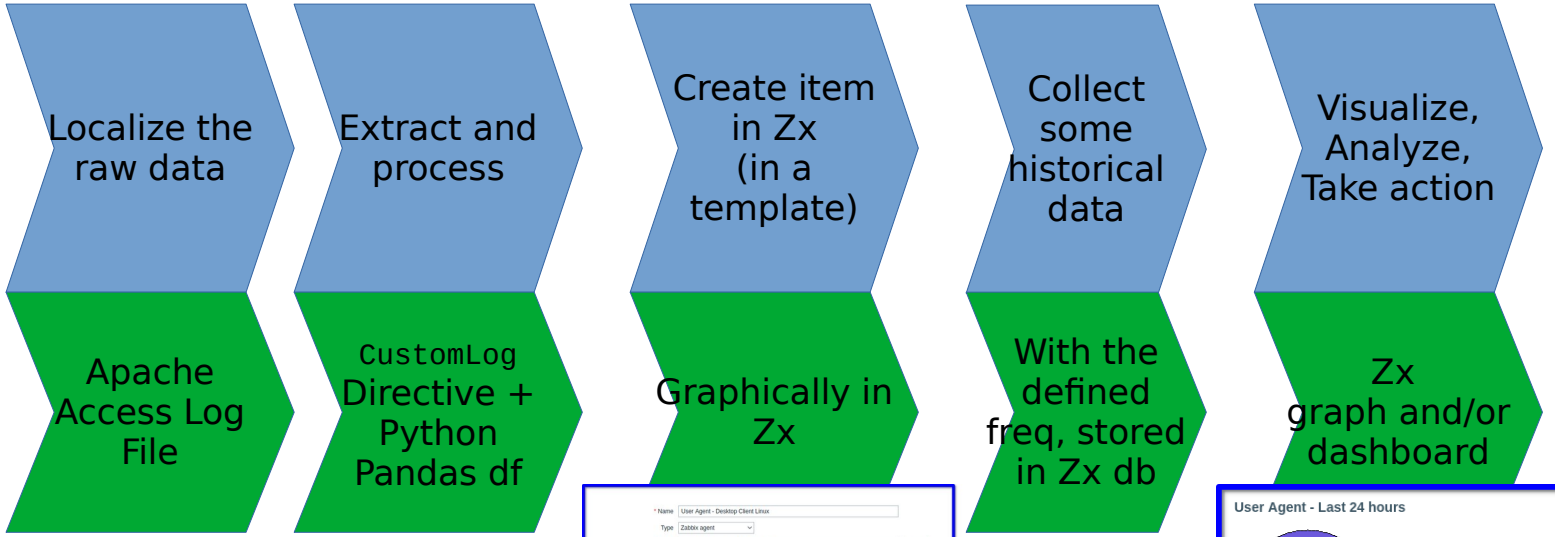
There are also templates (official or community) for container runtimes, Docker and LXD.



System Performance Dashboard for a LXD container

System Performance Dashboard included in the official "Linux by Zabbix Agent" Template

Sample Metric Pipeline: User-agent



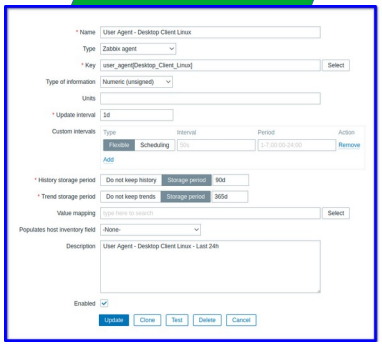

```
CustomLog access-csv.log
"\"%{%Y-%m-%d %T}t.%
{usec_frac}t\",\"%{User-
agent}i\""
```

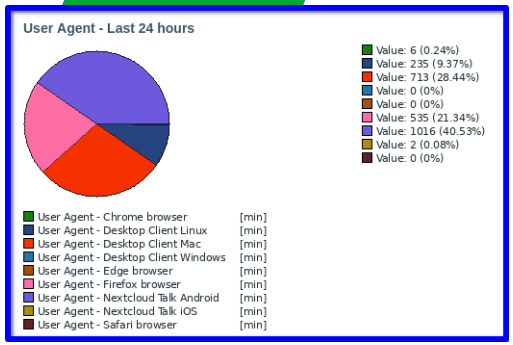
```
df_24h = df[df.timestamp>datetime.now() - timedelta(hours=24)]

df_24h["norm_user_agent"] = df_24h.user_agent.apply(get_norm_user_agent)

df_24h_agg = df_24h.value_counts('norm_user_agent')

json.dump(df_24h_agg.to_dict(),
        open("/var/lib/zabbix/output/monitor_apache2_user_agent.json","w"))
```

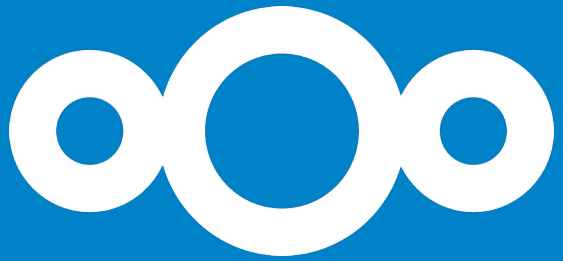




Conclusion

Conclusion

- We have seen a monitoring dashboard set for Nextcloud, basic in terms of metrics collected and components monitored
- For application monitoring, if the metric is available in serverinfo extract, query the DB or use the API
- Application monitoring is key for business reporting and decision-making
- If analytical queries are to be run on a regular basis, consider using a cold replica
- Come to our booth for any questions / chat, contact us or one of our partners for a review of your monitoring practice



Thanks!

pietro.marini@nextcloud.com