HANDLING HIGH VOLUMES IN LEGACY MONITORING SOLUTIONS

Nikola Valchanov

Abstract. *This paper discusses some capacity limits of known legacy processes for monitoring IT infrastructure. It describes known risks and respective mitigation techniques through automation or implementation of infrastructure monitoring platforms.*

Key words: infrastructure monitoring, data flow management.

1. Introduction

The process of monitoring and maintaining IT infrastructure is in constant evolution. This emergence of new tools and techniques is not dictated just by technology itself, but by the scale at which companies operate and the volumes of data that is stored and processed.

In many cases young product companies in their startup phase are hesitant to implement complex solutions that set them down to a certain infrastructure decision path. Instead, when inspecting the tooling of a startup with business traction one can find a high number of independent temporary solutions that are often accompanied by a process involving human operators.

This paper discusses how scale affects business processes where unstructured data is stored and then processed by employees. It considers risks involved and suggests mitigations by process automation or implementation of infrastructure monitoring platforms.

2. Storing and processing data of non-standard structure

Let us consider the following scenario – an understaffed startup is trying to implement a solution that collects a metric for in-house decision-making focused on a single customer.

The first thing a developer does is to dump the metric data in plain text on a media that does not require technical skills to operate so it's easily reachable. Commonly this is (but it is not limited to):

- A folder of text files containing free text dumps;
- A set of standard-format log files;
- A document in a folder published by a web server application;
- A mailbox.

All solutions above are simple and do not rely on a third-party platforms [1, 2, 4, 7] that need investments for deployment, ensuring high availability and performing maintenance. What is common across all those practices is that due to the lack of automation, stored data needs to be processed manually so that anomalies can be identified.

As the company grows and scale increases this solution requires more and more human resources to go through the stored data. The most common types of information stored in such scenarios are:

- System health information;
- System event information;
- Operational and/or performance metrics.

While most of the above are subject to automated processing, the practice dictates that these temporary solutions are often adopted as company standards. They are viable for small-sized infrastructure, thus tooling and processes are not replaced until a certain scale is reached.

3. Risks and mitigations in high-volume scenarios

There are several risks involved with the scenario above. Let's consider the general setup of the process represented on Figure 1.

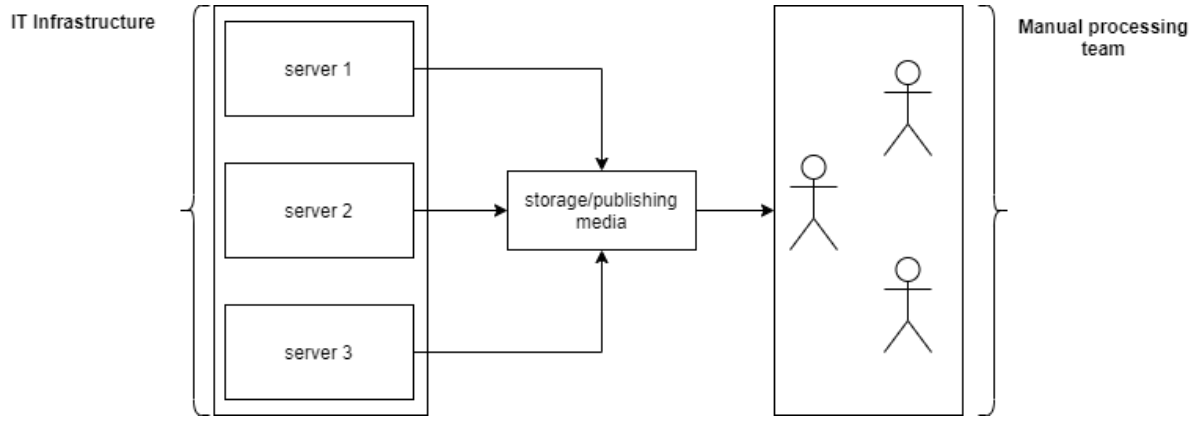

Figure 1. Manual processing of aggregated infrastructure data

In Figure 1 operational data is generated at the IT infrastructure and persisted in a storage/publishing media. Once stored data is then processed either reltime or at fixed time intervals by operators who interpret the data and escalate when issues are identified.

Risks

The process above poses several risks, where the obvious two are:

- Human error;
- Manual processing team management on bigger scale.

While human error is a very obvious risk it often is a risk to one customer. Once the customer base scales up, the processing team organization becomes a high-risk operation as the distribution of verifications each individual needs to process on daily basis needs to be scheduled and coordinated across the whole team. In this scenario the light-weight verifications effort quickly transforms into a heavy process that coordinates a very expensive team that is performing simple automatable verifications.

Risk mitigations

The obvious solution to the risk of human error is automation. There are two levels of automation that can be implemented to address the risks listed above:

- Automating the issue identification process;
- Automating the issue management process.

There are different approaches to issue identification, based on the type of the collected data:

- Timeseries based variance of a metric;
- Infrastructure system monitoring;
- Custom events monitoring.

In the case of a standard timeseries metric issue identification consists of building alert rules. Metric data can be stored in a database management system which is connected to a monitoring platform for building alerting rules.

In the case of infrastructure system monitoring the need for an additional layer in the tooling appears. This layer needs to collect infrastructure information such as capacity, load, availability etc. The third and

most complex case is monitoring custom events. To do that we need to store event information into a data source that is also compatible with the system monitoring tool so we can report on a combination of events monitored at different infrastructure level.

The second risk mitigation is the management of identified issues. This too is a process subject to automation. In practice, the majority of issues are of the same type – most commonly issues related to human error or contract breach. Companies usually have formalized checklists for handling the most common problems in their infrastructure and in most of the cases those are automatable. Automation of such processes can be orchestrated with infrastructure automation tool or a general purpose workflow engine. Based on the type of infrastructure issue the alerting mechanism can trigger a flow alongside the standard alert dispatch. Such flows can contact customers, trigger customer interaction interrupt or engage a tech/admin representative if counterpart investigation reaches a scenario outside of the automation scope.

4. Automation setup

Regardless of whether the solution is implemented through combination of different integratable systems or through a custom tool, the general setup of the automation is given on Figure 2.

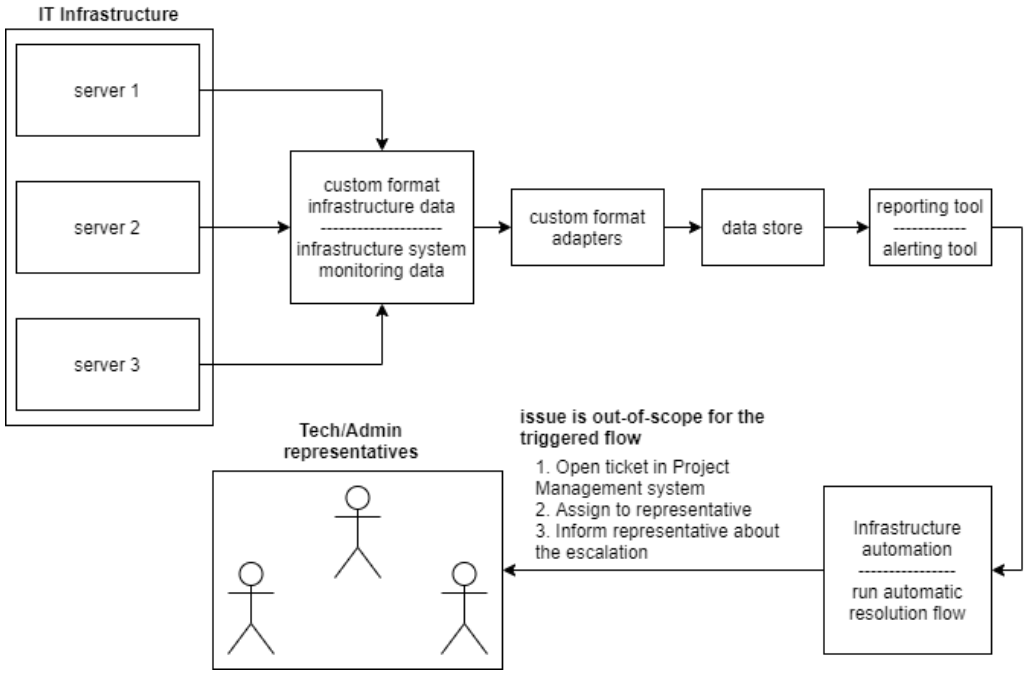

Figure 2. Automated processing of aggregated infrastructure data and alerting

The diagram on Figure 2 portrays the process of automated identification of issues, automated resolution attempt and automated initiation of the management process when issue is out-of-scope of the implemented automation. Let's consider a sample orchestration of tools that cover the requirements of the diagram above.

Timeseries metric information can be stored in most of the modern database management systems. Reporting and alerting are both features supported by the open source tool Grafana [2].

One of the most used tools in infrastructure monitoring currently is Prometheus [3], which can monitor capacity, load, availability and many more infrastructure metrics. Prometheus is also one of the supported data sources for Grafana through the PromQL [9] feature.

The third component requires custom event monitoring capabilities. The EventStoreDB [5] data source integrates with Prometheus through Prometheus exporters. This is how EventStoreDB can aggregate data from both Prometheus and custom sources and display, alert or trigger flows based on it through Grafana.

The last discussed automation is based on infrastructure automation or workflow management tools. Such workflows can be managed by tools such as SnapLogic [6]. Workflows defined in it can be triggered via a number of different protocols including HTTP which fits the supported actions by the other tools used in this setup.

5. Conclusion

The model for collecting, displaying, and managing event data in this paper covers the described cases of legacy event management solutions. It can be implemented either through a custom rule-based alerting tool or through a standard software stack of compatible tools that collect, store, and display data and alert based on user-defined rules.

The suggested implementation consists of scalable tools that can easily support high-volume data flows and large datasets.

References

[1] D. Berardi, F. Callegati, A. Melis, Sustainable Infrastructure Monitoring for Security-Oriented Purposes, GoodTechs '20: *Proceedings of the 6th EAI International Conference on Smart Objects and Technolo-*

gies for Social Good, (2020), 48–53, ISBN: 978-1-4503-7559-7, DOI: 10.1145/3411170.3411236.

[2] M. Chakraborty, A. Kundan, *Monitoring Cloud-Native Applications*, Apress, Berkeley, CA, (2021), ISBN: 978-1-4842-6887-2.

[3] M. Chakraborty, A. Kundan, *Prometheus*, Apress, Berkeley, CA, (2021), ISBN: 978-1-4842-6887-2.

[4] J. Hernantes, G. Gallardo, N. Serrano, IT Infrastructure-Monitoring Tools, *IEEE Software*, (2015), Vol. **32**, 88–93, ISSN: 0740-7459, DOI: 10.1109/MS.2015.96.

[5] `https://www.eventstore.com/eventstoredb`, (2021).

[6] `https://www.snaplogic.com/`, (2021).

[7] A. Komarek, J. Pavlik, L. Mercl, V. Sobeslav, *Metric Based Cloud Infrastructure Monitoring*, 3PGCIC 2017. Lecture Notes on Data Engineering and Communications Technologies, Springer, (2017), ISBN: 978-3-319-69834-2.

[8] Vit. Novotny, P. Sysel, J. Prinosil, Critical Infrastructure Monitoring System, *2021 IEEE 17th International Colloquium on Signal Processing & Its Applications (CSPA)*, (2021), ISBN: 978-1-6654-1484-5, DOI: 10.1109/CSPA52141.2021.9377303.

[9] N. Sabharwal, P. Pandey, *Working with Prometheus Query Language (PromQL)*, Apress, Berkeley, CA, (2020), ISBN: 978-1-4842-6215-3.

Nikola Valchanov[1,*]
[1] Paisii Hilendarski University of Plovdiv,
Faculty of Mathematics and Informatics,
236 Bulgaria Blvd., 4003 Plovdiv, Bulgaria
[*] `Corresponding author: nvalchanov@uni-plovdiv.bg`