PROJECT

Multi-tenant cybersecurity operations centre designed as an open cloud service with machine learning elements

Linked to work package KPB1 – Design of the virt. solution

Linked to output V1, the result of the analysis of available solutions

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1 Logging and monitoring in the cloud environment 3
1.1 Advanced logging strategies 3
1.1.1 Advanced monitoring techniques 4
1.1.2 Event Correlation and Analysis (ECA) 4
1.1.3 Security aspects of logging and monitoring 4
1.1.4 Automation 4
1.1.5 Alerting 4
2 OpenStack logging and monitoring 5
2.1 Logging in OpenStack 5
2.2 Monitoring in OpenStack 5
3 Openstack Charms from the monitoring perspective 6
4 Summary of the current state 6

Funded by the European Union
NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

1 LOGGING AND MONITORING IN THE CLOUD ENVIRONMENT

Logging and monitoring are an integral part of managing and maintaining cloud environments. These processes are critical for ensuring high performance, availability and security of cloud services.

Logging is the process of recording events, operations and errors that occur in a system. In a cloud environment, where services run on distributed systems, logging provides important information for the analysis and diagnosis of problems. It allows administrators and developers to track detailed operations of the entire system and to identify and quickly resolve problems that arise.

Monitoring is the process of continuously collecting, analysing and visualizing data on the performance and availability of cloud services. It makes it possible to obtain an overview of the state of infrastructure, applications and services in real time. This includes tracking resources such as CPU, memory, network activity and storage usage, as well as specific application and service metrics.

Key areas of importance of logging and monitoring:

- Security: Identification of security threats and weaknesses through the analysis of logs and monitoring data. The logs retain information about all operations performed and the users for whom they were performed
- Performance: Optimization of resource utilization and resolution of performance problems before they affect end users. By monitoring system metrics such as CPU, memory and disk usage and others, we are able to react in a timely manner in the event of their shortage or excessive use.
- Availability: Rapid diagnosis and remediation of service outages or failures. Recorded and stored logs constitute information about the entire system, and without access to them it is virtually impossible to determine what is happening/has happened to the system in order to perform a correct diagnosis.
- Compliance with legal standards: Adherence to legal and regulatory requirements concerning the recording and retention of data, from the perspective of so-called legal defensibility. It is necessary to maintain records of activity and accesses performed in the system, in case information needs to be retrieved retrospectively.

1.1 Advanced logging strategies

They include the aggregation, correlation and analysis of logs from various sources and services. Advanced techniques such as centralized logging platforms and logging agents enable efficient processing and storage of logged data. One well-known solution is the use of the ELK Stack (Elasticsearch for search and analysis, Logstash for log aggregation and transformation, and Kibana for visualization) or similar solutions that enable in-depth analysis and the extraction of useful information from huge volumes of logged data. [6]

1.1.1 Advanced monitoring techniques

They use a combination of passive and active monitoring to obtain a comprehensive overview of the state of the infrastructure and applications. Passive monitoring continuously tracks system metrics and performance indicators, while active monitoring involves generating and processing test transactions to measure response times and service availability. Solutions such as Prometheus, combined with Grafana for visualization, provide tools for collecting, storing and analysing metrics in real time. [6]

1.1.2 Event Correlation and Analysis (ECA)

It is an advanced technique within monitoring that enables the identification and automatic resolution of known problems through the correlation of various events and metrics. This method increases the efficiency of system management by minimizing false alarms (false positives) and enabling faster diagnosis and resolution of problems. Monitoring of metrics and logs can serve as an example. If we observe our system with enormous resource utilization and the logs show a large number of rejected attempts to access the system, we can assume that this is some kind of attack, and we are able to react more specifically and automatically to the correlation of these two events. [7]

1.1.3 Security aspects of logging and monitoring

They concern not only the capture and analysis of security incidents but also the protection of the logs and monitoring data themselves. The implementation of strong authentication and authorization mechanisms, encryption of data in transit and at rest, and regular auditing of access and activity are essential for protecting sensitive data and ensuring compliance with regulatory requirements.

1.1.4 Automation

It plays a critical role in logging and monitoring by enabling the automatic deployment of monitoring agents, the configuration of thresholds and alerts, or the execution of predefined actions based on the detection of specific events or metrics. The use of IaC (Infrastructure as Code) and configuration and management tools such as Ansible, Puppet, bash, etc. enables consistent and repeatable deployment and configuration of logging and monitoring services across the cloud infrastructure.

For effective logging and monitoring, it is important not only to select the right tools but also to define policies and procedures for log processing, the setting of alert thresholds and the automation of responses to identified problems. There is also a need to create alerting systems for automatic notification/response according to predefined rules.

1.1.5 Alerting

When collecting all metrics and logs from all modules, even if everything is located in a single centralized system, it is still very difficult to keep track of all the stored information, metrics and logs. There are alerting systems such as Prometheus Alerts or Grafana Alerts. Thanks to these systems, it is possible to set up rules that will be monitored on our behalf, and if they are exceeded, notifications can be configured to be sent to platforms such as Microsoft Teams or Discord. Notifications can be sent to any platform that supports web hooks. In Grafana Alerts, it is possible to create a rule. The rule is assigned a Querry that will retrieve current data from logs or metrics. These are compared with the configured threshold. If this threshold is exceeded, the alert transitions to a different state. The states available here are:

- Normal – the normal state of the monitored metric, or of the value from the Querry
- Pending – The monitored value has exceeded the configured threshold value at least once within a certain time
- Firing – The monitored value has not dropped below the threshold in the monitored intervals several times in a row

An alert in the Firing state thus passes to the Notification policy, in which the rules regarding the sending of Notifications are configured. These Alerting systems are essential for timely response, simplification and an overview of the entire monitored system without the need to look into this monitoring and logging system.

2 OpenStack logging and monitoring

OpenStack, as a comprehensive open-source CC platform, offers robust capabilities for the management and automation of large-scale virtual infrastructures. Effective logging and monitoring are critical for maintaining high performance, security and availability in OpenStack Clouds. These mechanisms enable operators to identify, diagnose and resolve problems while optimizing resource utilization.

2.1 Logging in OpenStack

Centralized logging is a fundamental requirement for the effective management of OpenStack deployments. Because OpenStack components are dispersed across various nodes, it is essential to record logs in centralized storage that enables log aggregation, search and analysis. Projects such as Elasticsearch, Logstash and Kibana (ELK) or Graylog are frequently used for their flexible log processing and visualization capabilities.

Logging levels in OpenStack can be set for different components, which allows detailed tracking of system activities or restriction of logging to critical events in order to reduce data volume. This flexibility is important for effective diagnostics and minimizing the impact on performance.

2.2 Monitoring in OpenStack

Monitoring of the performance and availability of OpenStack components is ensured through various tools and techniques. Projects such as Ceilometer and Monasca offer extensive capabilities for metric collection, resource monitoring and alert generation. These tools obtain data from various parts of OpenStack and support automated response to events in the system.

Ceilometer is used to collect resource usage data and metrics from OpenStack instances and services for the purposes of user billing, benchmarking and monitoring; it can record information on the usage of compute, storage and network resources, thereby providing a basis for performance analysis and resource optimization.

Monasca(Monitoring-as-a-service) is a highly scalable Multi-tenant monitoring platform for OpenStack that provides functions for collecting, storing, searching and visualizing metrics and logs. Monasca supports alerting and notification, enabling rapid identification and resolution of problems.

Logging and monitoring are an integral part of managing and maintaining OpenStack environments. The implementation of advanced logging and monitoring strategies increases transparency, enables rapid identification and resolution of problems, improves security and optimizes resource utilization. However, this requires careful selection of tools, configuration and continuous monitoring to ensure high standards of availability, performance and security.

3 Openstack Charms from the monitoring perspective

OpenStack Charms enables integration with popular monitoring and logging platforms such as Nagios, Zabbix and Prometheus, or the ELK Stack. For this integration, Juju charms have been developed that automate the entire deployment as well as the configuration of monitoring agents on OpenStack modules, along with the collection and export of metrics and logs to a central monitoring system. When a new OpenStack node or service is deployed via Charms, monitoring agents and the relevant configuration settings can also be added automatically as part of the process. This reduces the need for manual steps and simplifies keeping applications running and maintaining consistent monitoring across the entire Cloud infrastructure. An automated approach is among the most important properties in deploying and maintaining today's services.

In this deployment, it is necessary to bear in mind the dispersion of modules in a containerized environment, where some of the modules are deployed in containers for a certain consistency and isolation of the packages and tools they use. Modules such as Nova are deployed directly on physical nodes for direct access and allocation of resources to virtual devices. It is necessary to take into account the locations, whether of logging outputs, files, or when collecting metrics from specific lxc containers and nodes. Juju charms are intended to offer a better and simpler approach to deployment, to managing the entire life cycle, but also to fixing errors that arise, responding to events that occur, scaling, etc. Juju charms consist largely of Python code or bash scripts. The entire deployment must be predefined and programmed; every single hook or relation between individual components must likewise be implemented in Python charms. This gives rise to the need to implement the entire functionality, which means that if I want to deploy and manage an application using charms, I must define its configuration in code. If this application requires a connection to a database, this must be programmed. The same applies to monitoring. If the charm authors have not created this functionality for us, we are unable to perform it. Canonical also has a monitoring solution called COS, which we will discuss later.

4 Summary of the current state

Log information is available only on the specific machines on which the given OpenStack modules run and record their logs. These logs are not collected, maintained, backed up or analysed in any way. The same applies to metrics; however, some basic information on resource occupancy is available in the Horizon dashboard for OpenStack. This solution is not sufficient and serves only for informational purposes. There is only one way to determine the current and actual values of system metrics, namely by accessing the compute nodes and using Linux packages such as TOP and similar. From all the aspects mentioned, this state is inaccessible. It is necessary to analyse and build centralized access to all these logs and metrics.