TECHNOLOGY ACCEPTANCE MODEL FOR INCIDENTS MONITORING THROUGH CLOUD BASED PLATFORMS

Dr Manisha Divate, Assistant Professor
SVKM'S Usha Pravin Gandhi College of Arts, Science and Commerce
Vile Parle, Mumbai, India
divate.manisha.79@gmail.com

Krishnakumar Singh, Student
M.Sc. SVKM'S Usha Pravin Gandhi College of Arts, Science and Commerce
Vile Parle, Mumbai, India
singhkrishna73408@gmail.com

ABSTRACT

ServiceNow is a cloud-based platform for managing IT services, workflows, and processes more efficiently and effectively. ServiceNow has a centralized service desk for managing the incidents such as server failures, and applications issues. ServiceNow handles almost all kinds of IT, Cloud, Infrastructure Incidents, etc. Prometheus is open-source which has features such as real-time alerts and flexible query language for alerting event monitoring solutions. Prometheus takes help of an alert manager for generating alerts after crossing specific threshold values. This research work presents a technique to monitor service and usage checks through Prometheus. It uses Node-exporter as an agent to fetch the metrics that help to visualize data through visualization tools such as Grafana. Grafana is a data visualization tool that helps to display data into graphical representation.

The proposed model uses Prometheus with Alert-manager to integrate an operational monitoring system through ServiceNow to identify, diagnose, and resolve incidents in real time. ServiceNow provides a list of all Incidents from different tools for the operational team to handle real-time incidents and provide solutions. Quantitative based Technology Acceptance Model (TAM) is used to evaluate the system. The TAM framework helps to understand the user's acceptance level. The questionnaire is designed on the theoretical model of user satisfaction and ease of use, to evaluate the behaviour. It has been observed that users of the Incident Monitoring System are inclined to use the proposed system which makes their daily work simple.

Keywords:Technology Acceptance Model, Open-Source Cloud, Alert manager, Automotive Service Discovery, Incident Monitoring System.

Introduction

Earlier GitHub tools were used to manage cloud functionality and solve customer issues such as Incidents, Problems, and Tasks. This now uses Service to automate the cloud core functionality such as application availability, its performance & bug identification and solve customer issues of bugs by providing effective and needful solutions (Blackshaw ,Crawford, 2012) .

The challenge of monitoring the cloud performance, service availability, and Incident response management of a high-performance computing facility grows significantly as the facility employs and orchestrates more complex and heterogeneous systems and infrastructure.

However, more automation and integration are needed for collecting, aggregating, correlating, analysing, managing, and visualizing the problems present in the cloud infrastructure and for providing an effective and smooth service to the customers.

This research work presents a technique to monitor service and usage checks through an open-source tool, Prometheus. It uses Node-exporter as an agent to fetch the metrics that help to visualize data through Visualization tools such as Grafana. ServiceNow provides a list of all incidents from different tools for the operational team to handle real-time incidents and to provide solutions.

In IT infrastructure, ticketing tools play an important role in solving end user's queries who are connected remotely or within the infrastructure. Here the ticketing system helps to take user's query, process them and provide the required solutions(Gohil, Kumar, 2019). Figure 1 shows the Incident Monitoring System with automatic monitoring of tickets.

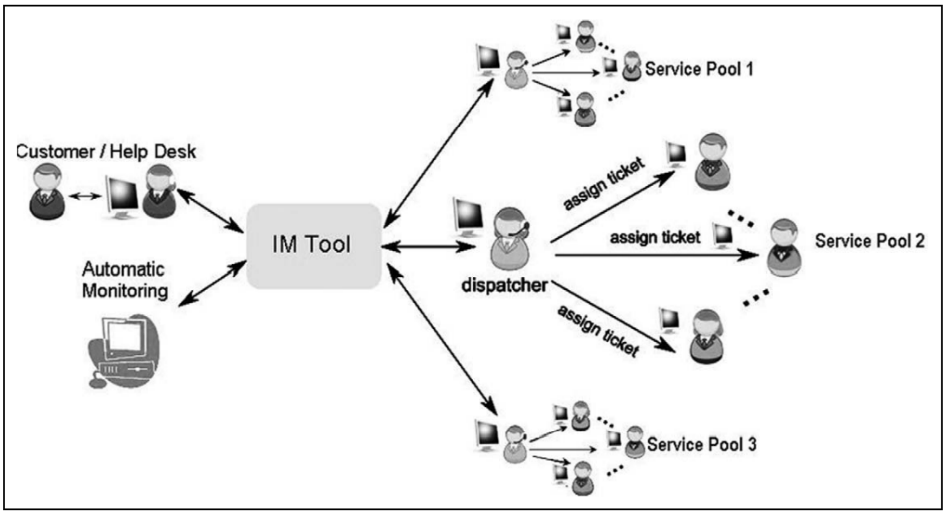

Figure 1. Automatic ticket management (Cavalcante V., 2013)

Here the ServiceNow tool is used for managing and handling tickets for integration. The most popular and similar tool is Jira. Table 1 shows the comparison of ServiceNow and Jira Service Management. Though the Jira Service Management tool is an efficient, and an useful emerging tool with an easy automation configuration system, most of the technologies such as Zabbix, Centreon and etc. already use ServiceNow for handling their problems, incidents, and tasks (Kotha, 2017).

Sr.no	Tools Comparison		
	Points	ServiceNow	Jira Service Management
1	Rank	3^{rd} Rank in Help Desk Softwares	2nd rank in Help Desk Software
2	Pricing	$100 user/month	Free, Standard, Premium, Enterprise (Free, $20, $40, etc.)
3	Analyses	Here analysis task comes with the user	Agent solves analyses issues
4	User Interface	Customized menu and tough to use	Modern UI and easy to use
5	Integration	Over 200 apps from 118 partners	Over 1000+ integrations
6	Used by	Many organizations already using this tool	New and adaptive upcoming tool

Table 1. Comparison between ServiceNow and Jira Service Management Tools

Challenges for ServiceNow and Prometheus User

ServiceNow is a popular IT Service Management (ITSM) software. It is used by thousands of businesses to manage their workflows, problems, service requests, and IT infrastructure. Prometheus, on the other hand, is an open-source event monitoring tool for real-time alerts. It gives businesses an overview of the state of their applications, enabling them to quickly identify and avoid problems (Kotha, 2017).

To maintain a high level of availability of all systems and services, organizations can use both Prometheus and ServiceNow to control and manage different aspects of their operations. If this research uses both systems, it's essential to connect them in order to diagnose and tackle issues before they impact the availability and performance of the applications. This proposed system uses a webhook. It is a call-back function used to simplify communication between two applications. It sends information using web sockets from one part of an app to another.

Alertmanager is an open-source platform that helps to send an alert to the destination system with the help of email, slack, webhook, and telegram, etc. Prometheus is an open-source time-series monitoring tool for sending alerts with the help of Alertmanager. Webhook sends information from Alertmanager to ServiceNow using web socket. Websocket helps to pass this information from the Alertmanager to the ServiceNow on the basis of severity such as critical and low. Here to make the communication secure and to eliminate the risk of data modification or snooping by malicious agents, SSL, self-signed certificate, is used.

Based on the rules/conditions set by the Alertmanager, system sends an alert. Prometheus uses to evaluate the metrics in every 10 to 15 minutes of intervals. Here metrics collect and store the data in time-series form. Metric records the information of cloud value with the timestamp. Alertmanager continuously checks the rules and generate the same type of alerts. This causes the duplication of the alerts in the queue. Therefore, to remove this de-duplicated of the alerts there is a need of filter or some mechanism.

Objectives of the Proposed System
The objectives of this study are
1. To provide an open-source solution to the problem of integration between ServiceNow and Prometheus through the secured channel of webhook in order to monitor the incidents.
2. To understand the user's attitude and behaviour towards the use of this system.

Existing systems use the tool but they are not in synchronization. Hence the integration of ServiceNow and Prometheus will be useful for IT users. This will be opensource cloud solution over the paid services such as Zigiwave.
Hence the system provides a single solution which integrates Prometheus with the ServiceNow in real time in order to detect and resolve the issues such as Request Items, Incidents, Task and Problem in the fastest way without visiting different tools on the portal. Set up seamless end-to-end data integration between both the tools to optimize cooperation between teams and help them to achieve more with less effort.

There are some of the questions which were unanswered
1. What are the possibilities of integrating Alertmanager with ServiceNow for an incident ticket generation?
2. What is the configuration required for Sending email from Alertmanager to ServiceNow which can convert it into the ticket?
3. Providing the specific severity based on "outage/issues" the ticket.
4. Automatic change of state of the ticket after the alert is generated.
5. Is there any mechanism to know the mapping of the alert with the ticket, in order to avoid the reduplication?

As discussed in the above section, integration of ServiceNow with Prometheus for generating alerts is the requirement for using a single pane. The tickets should be generated based on defined rules and the severity of the problems. For example, for server's CPU usage check, we can define percentage threshold like 75% for "High" and 90% for "Critical" severity.

Reduplication problem can be solved by adding the interval period between ticket generation. For example, if the interval of 1 hr is set and if no tickets got triggered with this problem so it will send the ticket once and keep a track. If any tickets for the same issue get generated during this time, then it will get captured by payload cluster and will be grouped and send after completion of desired time interval i.e., 1 hours.

On this basis the questionnaire is designed which contains personal information, perceived usefulness, perceived ease of use, user satisfaction (Chuttur, M.,2009).

Benefits of using Proposed System
During this research work, to reduce the human leverage task, the system is automated to synchronize the ITSM role and monitoring.

The proposed system helps to create an efficient single pane for monitoring the incidents from different sources. This gives real-time experience to handle live Incidents. Also, this automation will truly help when the servers and instance count will increase as in manual process managing is difficult by visiting different portals or applications

Methodology: Steps and procedure for Integration
These are the steps needed to be done to achieve Integration between ServiceNow and Prometheus for Incidents and problem tickets get generated.

Step1. Some pre-requisite requires to be there on the machine so these steps to be done that are:

1. Installation of the Prometheus tool on the server.
2. ServiceNow tools with admin/Editor privilege.

3. Alert manager should be installed on the same server where Prometheus is deployed
4. Check for its service running
5. Firewall port should be opened for intercommunication between the source and the destination server.
6. Self-signed SSL certificate required for secure communication between the application using web sockets.

Integrate the Prometheus open-source monitoring tool with an Event Management by adding a standard webhook in the Prometheus Alertmanager. Webhook is a way to feed the messages from the different applications to the system. This webhook will receive the alerts from Alertmanager which is generated in the Prometheus and sends the detail of the alerts.

Step2: Pre-requisites for ServiceNow for mapping Alerts:
Ensure that the Event Management Connectors (sn_em_connector) plugin is installed on the ServiceNow Platform instance.

Verify the Configuration Item (CI) for the host managed by Prometheus exist in ServiceNow. CI is the specific name for the server that ServiceNow has been stored. The Configuration Management Database (CMDB) is the ServiceNow database which stores information about all the services. CMDB stores the support information for each service offering in a Configuration Item (CI) specific to that service. Whenever alerts are triggered in the payload Virtual Machine (VM), which stores the data of alerts triggered by Alertmanager, and are send to the ServiceNow. Here ServiceNow first checks the presence of CI in CMDB. If CI exists in CMDB then only the alerts get mapped and accepted by ServiceNow. ServiceNow discovers the alerts generated by user or production environment with the help of CI.

Role required: evt_mgmt_integration
E.g.: for prod1: os_prod1_central_mon
E.g.: for prod2: os_prod2_central_mon
This CI is used to help the Prometheus tool to discover the production environment and its physical server containing the error. This helps the users to rectify easily identify the problem and provide quick solutions.

Step3: Procedure for Integration of Prometheus and Alertmanager

1. In the Prometheus instance, create an alert rule.
   a. Create a file with an extension of (.rules) for the alert rules with permission of sudo chmod 774 <filename>. Edit the file Prometheus.yml that exits under "/etc/prometheus/" following location. Add the rule file path in the Prometheus.yml file
   b. Create the alert rule as per the Prometheus format using PromQL language

2. In the Prometheus instance, add a webhook:
   a. Navigate to Alertmanager .yml and add a webhook entry in the receivers.
   b. Enter the endpoint URL in receiver's section. For example, E.g.:http://53.137.10.3:8085/alerts/OpenstackAlertmanager /addAlert
   c. Type basic_auth below http_config
   d. For a secure communication between applications, add SSL with the webhook that uses WebSocket to pass the information. For adding SSL in the webhook, create CSR i.e. Certificate Signed Request which is specially used to send the message in an encrypted format.
   e. Add a username and a password in basic_auth on /etc/alertmager, then add the webhook name created in earlier step in the receivers /route
   f. In inhibit_rules, define matching condition for an alert to get triggered and the details which need to be received by the webhook based on severity of an alert.
   g. Restart the Prometheus to reflect the changes.
   h. Restart the Alertmanager to reflect the changes.

3. The mechanism behind the Prometheus Alertmanager and ServiceNow.
   The Payload VM is created with multiple load balancer for managing the incident tickets queues. Payload VM consist of Java code function which collects the XML data triggered by the Alertmanager. This XML data gets encrypted before sending to the ServiceNow. This encrypted data then send to the ServiceNow CI. Here CI will check for the CI item's existence in the CMDB. If CI Item is found, it

maps the XML data to a table form otherwise it fails to the send data to the ServiceNow. In this case ticket will not be generated.

The above steps ensure the integration of Prometheus with the ServiceNow in order to monitor the incidents in a real-time.

Data Analysis and Result

Technology Acceptance Model (TAM) is a theoretical framework introduced by Fred Davis predicts the acceptances of a user for a new technology. The aim of the research is to develop an Incident Monitoring System and check its acceptability and the user satisfaction level (AbuDalbouh, 2013; Lee, 2003; Liu, 2014)

According to TAM, following are the parameters to be consider to evaluate the user's satisfaction.

A. **Perceived of usefulness satisfaction**
   a. The proposed system will help to track the incidents, problems and task records into the ServiceNow Portal and get the information quickly.
   b. The proposed system of tracking records will help to get the list of all the issues in a single ServiceNow portal
   c. The proposed system will help to save the time and get the progress report quickly
   d. The open-source cloud solution will save the time of operational team for issue tracking

B. **Perceived Ease of Use**
   a. Learning to work on the ServiceNow portal for incident tracking would be ease for me.
   b. My interaction with open-source cloud solution for tracking the incidents, problems and tasks progress would be clear and understandable.
   c. I would find an open-source cloud solution for tracking on incidents, problems and tasks progress more flexible to interact
   d. It is easy and user-friendly for me to become skilful at using the ServiceNow portal to track the ticket progress.

C. **User Satisfaction**
   a. I am completely satisfied in using open-source cloud solution for tracking incidents, problems and tasks progress.
   b. I found it helpful in getting details of incident, problems and tasks of the ticket description.
   c. I am able to get records of the issues rapidly.
   d. I believe that by using this open-source solution for tracking issues on server/ cloud will increase the quality of the service.

These are the questions given to the user in order to understand their acceptance level and user satisfaction level for the system (Masrom, M.,2007). They are asked to fill the responses for the questions in 5-scale Likert scale (1: poorly acceptable; 5: Highly acceptable). Table 2 and 3 shows the usefulness and satisfaction of accessing the system.

	Questions	**1: Poorly acceptable** **2: Fairly acceptable** **3: acceptable**	**4: Moderately acceptable**	**5: Highly acceptable**
Perceived of usefulness satisfaction	The proposed system will help to track the incidents, problems and task records into the ServiceNow Portal and get the information quickly.	0	3	12
	The proposed system of tracking records will help to get the list of all the issues in a single ServiceNow portal	0	0	15
	The proposed system will help to save the time and get the progress report quickly	0	6	9
	The open-source cloud solution will save the time of operational team for issue tracking	0	0	15

Perceived Ease of Use	Learning to work on the ServiceNow portal for incident tracking would be ease for me.	0	0	15
	My interaction with open-source cloud solution for tracking the incidents, problems and tasks progress would be clear and understandable	0	12	3
	I would find an open-source cloud solution for tracking on incidents, problems and tasks progress more flexible to interact	0	0	15
	It is easy and user-friendly for me to become skilful at using ServiceNow portal to track the ticket progress.	0	0	15

Table 2. Likert responses of the user for usefulness and ease of use of system

	Questions	**1: Not Satisfied** **2. Slightly Satisfied** **3. Satisfied**	**4: Very Satisfied**	**5. Extremely Satisfied**
User Satisfaction	I am completely satisfied in using open-source cloud solution for tracking incidents, problems and tasks progress.	0	3	12
	I found it helpful in getting details of incident, problems and tasks of the ticket description.	0	0	15
	I am able to get records of the issues rapidly.	0	9	6
	I believe that by using this open-source solution for tracking issues on server/ cloud will increase the quality of the service.	0	0	15

Table 3. Likert responses for the user's satisfaction

Table 2 and 3 shows that the users are satisfied with the use of the system as it solves the problems quickly and user can easily keep the tracks of the tokens. User uses the single portal for all kinds of tickets.

Conclusion

The proposed system integrates the ServiceNow and the Prometheus in order to monitor the incidents. The system suggests the steps to configure the Prometheus. Prometheus has the rule file which contains the condition for triggering an alert. The alerts then passed to the Alertmanager. Alertmanager uses the webhook to send a secure and encrypted information to the Payload VM.

Reduplication of similar kind of alerts in a queue can be resisted by adding the interval inside the payload VM and by keeping the alerts in the queue till the interval period gets over. Automatic incident resolution process needs to be added in order to resolve an incident by its own.

From the TAM, it is concluded that the proposed system will help the operation team to make the resolution of an incident automatically with easy and single pane portal solution.

References

AbuDalbouh, H. (2013). A questionnaire approach based on the technology acceptance model for mobile tracking on patient progress applications. Journal of Computer Science, 9(6), 763–770.

Blackshaw, T., Crawford, G. (2012). Event Management. The SAGE Dictionary of Leisure Studies. https://doi.org/10.4135/9781446213278.n80

Cavalcante, V., Pinhanez, C., de Paula, R., Andrade, C., de Souza, C., & Appel, A. (2013). Data-Driven Analytical Tools for Characterization of Productivity and Service Quality Issues in IT Service Factories. Journal of Service Research, 16(3), 295–310.

Chuttur, M. (2009). Overview of the technology acceptance model: Origins, developments and future directions.

Gohil, F., Kumar, V. (2019). Ticketing System. International Journal of Trend in Scientific Research and Development, Volume-3(Issue-4), 155–156.

Kotha, V. (2017). Customer-Centric Service Management Using Servicenow.

Lee, Y., Kozar, K. A., & Larsen, K. R. (2003). The technology acceptance model: Past, present, and future. Communications of the Association for information systems, 12(1), 50.

Liu, Z. Y. (2014). An Analysis of Technology Acceptance Model. September 2014, 1–14.

Masrom, M. (2007). Technology acceptance model and e-learning. Technology, 21(24), 81.

Marangunić, N., Granić, A. (2015). Technology acceptance model: a literature review from 1986 to 2013. Universal access in the information society, 14, 81-95.

Sabharwal, N., Pandey, P., Sabharwal, N., & Pandey, P. (2020). Getting started with prometheus and alert manager. Monitoring Microservices and Containerized Applications: Deployment, Configuration, and Best Practices for Prometheus and Alert Manager, 43-83.

Sukhija, N., Bautista, E., James, O., Gens, D., Deng, S., Lam, Y., & Lalli, B. (2020, November). Event management and monitoring framework for HPC environments using ServiceNow and Prometheus. In Proceedings of the 12th international conference on management of digital ecosystems (pp. 149-156).