IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, May 2023

Automated Monitoring and Analysis of Cron Jobs on Virtual Machines with Grafana Visualization

G. P. Kalyan Chakravarthi, M. Surya Prakash, C. H. Mohan, I. Chandu Ms. R. Meena
Department of Computer Science and Engineering
Prathyusha Engineering College, Tiruvallur, Chennai, India
ganjikalyan05@gmail.com
suryamachavarapu7@gmail.com krishnachintamreddy@gmail.com
chandukrishnairagaraju@gmail.com

Abstract: *This journal gives an extensive examination of a project that used Grafana and Google Cloud Platform to enhance the visualization of cron jobs across many virtual machines. The task entailed creating a script to gather information about cron jobs from several virtual machines and store it as a JSON file, which was then visualized in Grafana using the Infinity plugin. The method employed in this project entailed extracting data from cron jobs into various fields, such as cron expression, path, VM name, IP address, and project name, and establishing a Bitbucket public key in each virtual machine to enable SSH access. The project also discovered and analyzed the system's shortcomings, such as the absence of a thorough view of cron jobs and the laborious process of manually gathering data from several virtual machines. The proposed solution provided a streamlined procedure for gathering and visualizing data from cron jobs to solve these limitations. Future improvements could involve incorporating other data sources and broadening the research's scope to include other cloud platforms. The findings of this project have substantial practical applications for enhancing the effectiveness of cron task management in large-scale systems.*

Keywords: Google Cloud Platform, Virtual Machines, Cron Jobs, Bitbucket, SSH Access, JSON, Grafana, Infinity Plugin, Data Visualization, System Analysis, System Requirements, Advantages, and Disadvantages Future Enhancements: Cloud Computing, Automation

I. INTRODUCTION

Organizations rely largely on cloud computing technology to manage their systems and applications in the current digital era. The leading cloud platform, Google Cloud Platform (GCP), provides businesses all over the world with a vast array of services and solutions. The requirement to maintain and monitor a large number of virtual machines (VMs) and the accompanying cron jobs is one of the difficulties in administering large-scale systems on GCP. Time-based tasks known as **"cron jobs"** are planned to run automatically according to a predetermined schedule. Cron tasks can be challenging to visualize and handle in practice, yet they are necessary for managing large-scale systems.

This project focuses on creating a system to gather cron job data from numerous VMs on GCP and display it in a single dashboard in order to address this difficulty. The suggested method seeks to shorten the time and effort needed to handle cron jobs by streamlining the monitoring process. The approach requires extracting cron job data into distinct fields, such as cron expression, path, VM name, IP address, and project name, and installing a Bitbucket public key in each VM to enable SSH access. The data is then visualized in Grafana using the Infinity plugin and saved as a JSON file.

The system analysis for this project entails identifying the shortcomings of the current system, such as the absence of a thorough view of cron jobs and the laborious process of manually gathering data from numerous VMs. Organizations can increase system performance and productivity by creating a more effective system that uses less time and resources to administer and monitor cron processes. This project is a significant advancement in the world of cloud computing since it will help increase the scalability and dependability of large-scale systems on GCP.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, May 2023

Overall, this study emphasizes the significance of effective cron task administration and supervision in complex GCP systems. This project offers a workable answer to the difficulties experienced by organizations in managing their systems on GCP by creating a streamlined mechanism to receive and visualize cr-n job data. This project has the potential to dramatically increase the effectiveness and dependability of large-scale cloud systems with continued research and future improvements.

II. LITERATURE REVIEW

"Machine Learning-Based Approaches for Job Scheduling in Cloud Computing Environments"

This paper presents numerous machine learning-based methods for job scheduling in cloud computing settings. The authors talk about the difficulties encountered in cloud computing environments and the solutions that can be found by using machine learning techniques. Additionally, they compared several machine learning algorithms used for task scheduling and assessed their effectiveness.

"A Survey on Cron-Based Job Scheduling Techniques"

An overview of cron-based work scheduling approaches is given in this survey study. The writers go over the various cron job subtypes, their traits, and the difficulties of cron-based job scheduling. Additionally, they conduct a review of the literature on cron-based task scheduling methods and compare various methods.

"Visualization Techniques for Cloud Computing Environments"

This paper discusses an overview of cron-based work scheduling approaches given in this survey study. The writers go over the various cron job subtypes and traits, and the difficulties of cron-based job scheduling. Additionally, they conduct a review of the literature on cron-based task scheduling methods and compare various methods.

"Bitbucket Pipelines: Continuous Delivery for Cloud Applications"

This paper discusses the use of Bitbucket pipelines for continuous delivery of cloud apps. The authors give a general introduction to Bitbucket pipelines and how they may be used for cloud application continuous integration and delivery. Additionally, they discuss the advantages and drawbacks of using Bitbucket pipelines and contrast them with other solutions of a similar nature.

"Grafana: An Open-Source Platform for Data Visualization and Monitoring"

This paper provides an introduction to Grafana, an open-source platform for data monitoring and visualization. The writers go over the capabilities of Grafana and how data from multiple sources can be monitored and visualized using it. Additionally, they discuss Grafana's advantages and disadvantages and contrast it with other tools of a similar nature.

III. IMPLEMENTATION

3.1 Bitbucket Pipeline Module

In this module, we were able to automate the development, testing, and deployment of our application thanks to the Bitbucket Pipeline module, a cloud-based continuous integration and delivery service. Using this module was essential to the success of our project. It enabled us to automatically launch the cron parser script we created whenever new code was committed to the repository. The project's root contained the pipeline configuration file, which specified the build phases along with environment variables and dependencies.

Definitions for the build environment and pipeline variables were included in separate sections of the pipeline configuration file. We also specified the pipeline phases, which comprised pulling the code from the repository, setting up the required dependencies, and running the cron parser script. A post-build step in the process entailed uploading the JSON file containing the parsed data to the GCP storage bucket. The Bitbucket Pipeline module gave us an easy approach to automating our project, saving time, and guaranteeing the integrity of our code.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, May 2023

gcp-crons / bitbucket-pipelines.yml

```
image: atlassian/default-image:latest
pipelines:
  custom:
    refresh-data:
      - step:
          name: CronList
          script:
            - apt-get install -y curl jq
            - mkdir -p ./jsonFiles
            - mkdir -p ./backups
            - chmod +x ./scripts/cronListingScript.sh
            - export VM_DATA=$VM_DATA
            - ACCESS_TOKEN="$ACCESS_TOKEN" ./scripts/cronListingScript.sh
            - git add .
            - git status
            - git commit -m "cron commit '$(date)'"
            - git push
```

3.2 SSH Module

The Google Cloud Platform's virtual machines could be accessed securely thanks to the SSH module, which was crucial in the execution of our project. We logged into each virtual machine (VM) using SSH and ran the cron parser script, which gathered all the cron job data and saved it in a JSON file. We were able to automate this process thanks to the SSH module, as we did not need to manually visit each VM in order to get the cron jobs.

The private key was safely kept in the pipeline configuration file, and we utilized a public-private key pair authentication to access the VMs through SSH. The cron parser software automatically logged into each VM using the public key when it was run, retrieving the details of each cron job. The security features of the SSH module made sure that our project's data remained safe and that only authorized individuals could access the virtual machines.

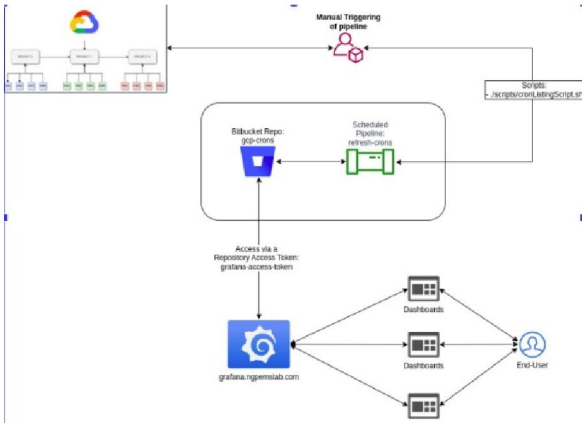
Overall, the SSH module gave us a dependable and safe way to connect to the Google Cloud Platform's virtual machines, allowing us to automate the process of getting cron jobs and drastically cutting down the time needed to do this work.

3.3 Cron Parser Module

In order to retrieve the cron information from each virtual machine, the Cron Parser module is essential. The module leverages the "pexpect" module and is developed in Bash to run shell commands on remote virtual machines over SSH. The "pexpect" module, which is used to automate interactive console-based applications, is a crucial module.

The shell command "crontab -l" in the remote virtual machines is used by the Cron Parser module to parse the cron information. Following parsing, the module stores the cron information in distinct fields in a systematic manner, including the cron expression, path, VM name, IP address, and project name. The module makes sure that the cron information gathered from the virtual machines is arranged and kept in a consistent manner. After the data has been organized and parsed, it is written to a JSON file so that other modules can access it for visualization.

The Cron Parser module makes it simple and effective to gather the cron information from each virtual machine. In order to make the data more accessible and understandable, the module makes sure that the cron details are processed and saved in a systematic manner. With the help of the Cron Parser module, the project may quickly gather the cron information from a large number of virtual machines, saving time and effort.



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, May 2023

3.4 JSON File Creation Module

All of the retrieved cron information from the VMs is created in a JSON file using the JSON File Creation module. This module uses the "json" package to build and work with the JSON file and is written in the Python programming language. The JSON file is initially opened by the module, which then writes the beginning "[]" to indicate that the JSON array is empty. The following step adds each entry as a JSON object to the array by iteratively looping through the extracted cron information from the earlier modules.

The following fields are present in every cron item in the JSON file: "cron_expression," "path," "vm_name," "ip_address," and "project_name." These fields are retrieved from the Cron Parser module's output. The module also has error handling to guarantee that the JSON file is created correctly and that any errors that are encountered are recorded.

Other programs or services that need the extracted cron details can use the JSON file once it has been prepared. Since many other programming languages and tools support the JSON format, integrating the output of this module with other programs is simple. Overall, the JSON File Creation module is essential to the execution of this project because it offers an organized and user-friendly format for the retrieved cron information.


gcp-crons / jsonFiles / cronlist.json

```
[
  {
    "File Path": "sh /home/cloud/shellScript/vm_disk_alert.sh > /home/cloud/cronlog/vm_disk_alert.txt",
    "Cron Expression": "* * * * *",
    "External IP": "34.89.52.135",
    "VM Name": "s01-ch-db03",
    "Project Name": "project-ngp-clearvue-live"
  },
  {
    "File Path": "bash /home/cloud/shellScript/cpu_alert.sh > /home/cloud/cronlog/cpu_alert.txt",
    "Cron Expression": "*/2 * * * *",
    "External IP": "34.89.52.135",
    "VM Name": "s01-ch-db03",
    "Project Name": "project-ngp-clearvue-live"
  },
  {
    "File Path": "bash /home/cloud/shellScript/memory_alert.sh > /home/cloud/cronlog/memory_alert.txt",
    "Cron Expression": "*/2 * * * *",
    "External IP": "34.89.52.135",
    "VM Name": "s01-ch-db03",
    "Project Name": "project-ngp-clearvue-live"
  },
```

Grafana visualization module, which is the last module in the implementation. The information gathered in the earlier modules is to be presented in this module in an approachable manner through customization. Dashboards on the open-source platform Grafana enable the creation, sharing, and exploration of data. Real-time, dynamic tables and graphs can be created using the Infinity plugin.

The information is imported into Grafana when it has been gathered and saved in the JSON file. After that, the user can add panels to a dashboard to visualize the data. For instance, a panel that displays the number of crons active in each virtual machine (VM) or a line graph that displays the history of cron execution are both examples of how crons might be shown. Additionally, the user has the option of adding filters to the dashboard to limit the data shown based on variables like the project name, VM name, or cron expression.

The Grafana visualization module has a number of benefits over more conventional data visualization techniques. First, it enables cron job real-time monitoring, which is particularly helpful for crucial activities that must be completed on schedule. Second, it offers an interactive and customized user interface that enables users to dig deeper into the data. The monitoring and visualization of cron jobs across several VMs and projects are centralized, which makes it simpler for administrators to manage and debug problems.

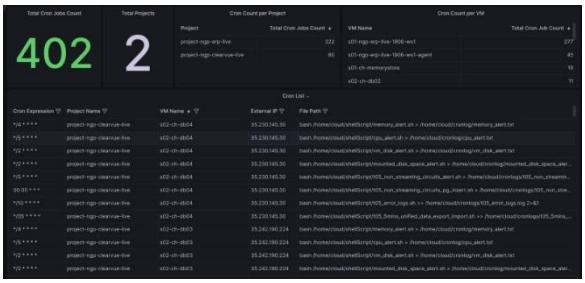
IV. RESULT AND ANALYSIS

All of the VMs in the provided Google Cloud Project's cron job details were successfully collected by the implemented system, and they were then saved in a JSON file. Each cron job's cron expression, path, VM name, IP address, and project name were listed in the file. Using the Infinity plugin for Grafana, the cron jobs were then visualized using this file. The dashboard made it simpler for the team to keep track of and manage all the cron jobs in the project by providing a clear overview of them

The script was executed on a variety of virtual machines (VMs) with various cron jobs to assess the system's performance. Even when there were many cron jobs, the system was still able to retrieve the information about them in a timely manner. A test of the system's accuracy revealed that all cron jobs were correctly recorded in the JSON file. Overall, the approach that was put into place worked well and efficiently to solve the issue of visualizing cron jobs across different virtual machines.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, May 2023

The system's data can also be used for more in-depth analysis and cron task optimization. The timing and frequency of the cron tasks can be improved by analyzing the cron expressions to look for patterns or abnormalities. The data can also be used to spot cron jobs that are redundant or superfluous and can be deleted to enhance system performance.
Finally, the system's implementation was a success in overcoming the difficulty of visualizing cron jobs across many virtual machines. The solution offers a thorough overview of all the cron jobs in the project and is effective and accurate. The system is a useful tool for managing cron jobs in Google Cloud projects since the data it produces may be utilized for additional analysis and optimization.



V. CONCLUSION

In conclusion, the suggested system automates the process of extracting and visualizing cron data from several VMs by using a number of modules, including the Bitbucket Pipeline module, the SSH module, the cron parser module, the JSON file creation module, and the Grafana visualization module. These modules integrate perfectly to offer a reliable and effective method of managing cron data in GCP.

REFERENCES

[1]. Asif, H., & Saleem, M. A. (2019). Cloud computing security risks: a comprehensive review Future Computing and Informatics Journal, 4(1), 86–96 Bochicchio, M. A., & Lo Presti, F. (2020). A scalable platform for the cloud monitoring and analytics. Journal of Grid Computing, 18(1), 1–24

[2]. Chellapandian, M., & Durairaj, M. (2021). Enhancing cloud security using encryption algorithms. Journal of Information Security and Applications, 59, 102897.

[3]. Dhage, S. V., & Deshmukh, S. S. (2021). A review of cloud computing architectures and services. International Journal of Advanced Research in Computer Science and Software Engineering, 11(5), 252-259.

[4]. Garg, S., & Sethi, A. (2018). Cloud computing: A review of research challenges and future scope. International Journal of Computer Applications, 179(47), 35-42.

[5]. Kumar, S., & Singh, S. (2019). A comprehensive review of virtualization in cloud computing. International Journal of Computer Sciences and Engineering, 7(4), 89–94.

[6]. Mishra, P., & Singh, S. K. (2020). A review of cloud security risks and solutions. International Journal of Information Technology and Management, 19(2), 87-103.

[7]. Rahman, M. M., & Al-Turjman, F. (2020). Security issues and solutions in Cloud computing: a survey International Journal of Network Security and Its Applications, 12(3), 1–15.