IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Automation of Linux administrative tasks using BASH shell scripts

Jeyadev Needhi, Sangeetha RK, Swathy KS, Latha Sri SA, and Priyadarshni.V
Department of Computer Technology
Madras Institute of Technology,Anna University Chrompet,Chennai, India

***Abstract*—**Linux system administrators are productive when they create the shell programs that automate their tasks.The more automation we have in place, the more time we have available to fix real problems when they occur and to contemplate how to automate even more than we already have. Instead of typing the same group of commands on the command line, it is better to automate it by adding them in a script. Linuxserver automation is the process of automating the provisioning, patching, scheduling, security, compliance and other linux based tasks across virtual and physical servers. Most large enterprises employ linux system infrastructure. To efficiently manage them, IT administrators must monitor, update and secure the servers to ensure that they are operating at their best. Linux does not have a built-in backup tool which adds complexity to the backup process.As a result linux administrators need to customize their linux infrastructure initially to fit their environment.Though Linuxautomation is achieved by using bash shell script in most of the cases we attempt to enhance the process of automation by scheduling the CRON job with the help of an open source tool, ANSIBLE.The vision of our project is to automate the following tasks

***Index Terms*—**Automation,Bourne Again Shell,Ansible,Cron daemon,Threshold memory, Job scheduler

I. INTRODUCTION

Linux is a free and open-source operating system that is widely used in server environments, cloud computing, and embedded systems. It was first created by Linus Torvalds in 1991 and has since grown into a powerful and flexible operating system with many different distributions, such as Ubuntu, Debian, Fedora, and CentOS.One of the key features of Linux is its modularity and flexibility. It can be customized and configured to suit a wide range of applications and use cases, making it a popular choice for developers, system administrators, and other IT professionals. Linux also has a robust command-line interface and a vast array of command- line tools and utilities, which can make it very powerful for system administration tasks.Another key strength of Linux is its security features. Linux systems are inherently more secure than other operating systems due to their strong user and file permission management, and the availability of many security- focused tools and utilities. Additionally, the open-source na- ture of Linux means that vulnerabilities can be discovered and fixed quickly by the community, making it a very resilient and secure operating system.Since linux is a powerful and flexible operating system that has many advantages for use in a wide range of environments it is essential to optimize the working environment in linux using various other tools and softwares. In general linux users tend to have more technical expertise and experience with computer systems, as linux is often used in professional and technical settings. Windows users, on the other hand, are more likely to be casual or general computer users who may not have as much technical knowledge. Linux users tend to value the customization and flexibility of the operating system, and often prefer open- source software that they can modify and customize to suit their needs. Windows users may be more interested in ease of use and pre-packaged software solutions.As mentioned in [1] linux is often seen as more secure than Windows or Mac OS X, but this does not mean that there are no security concerns to be had when running it. Attackers can crack simple passwords over a network, vulnerabilities can be exploited if firewalls do not close enough ports, and malware can be downloaded and run on a Linux system. In addition, sensitive information can be accessed through physical or network access if proper permissions are not set on the files or directories containing it. Linux automation refers to the process of automating various tasks and operations on a Linux system using various tools and techniques. This can help improve efficiency, reduce errors, and save time by automating repetitive tasks that would otherwise require manual intervention. There are several tools available for Linux automation, such as shell scripts, cron jobs, Ansible, Puppet, and Chef, among others. These tools allows us to automate tasks such as system backups, software installa- tion and updates, log rotation, security scans, and more. Linux automation is widely used in system administration, DevOps, and cloud computing, as it allows organizations to manage large-scale infrastructures more efficiently and effectively. By automating routine tasks, system administrators and developerscan focus on more critical tasks and projects, leading to greater productivity and better outcomes.Automation can help to im- prove efficiency, reduce errors, increases security and simplify the management of complex systems. Linux automation refersto the use of scripts, tools, and other techniques to automate the configuration, deployment, and management of linux systems.

II. LITERATURE SURVEY

The paper [2] ”Study of the Dirty Copy on Write, a Linux Kernel memory allocation vulnerability” presents an analysis of the Dirty Copy on Write (Dirty COW) vulnerability in the Linux kernel memory allocation system. [2] provides

a comprehensive literature survey of the previous research conducted on the Dirty COW vulnerability and highlights the key findings and limitations of the existing studies. [2] also proposes a new method to detect and prevent the Dirty COW vulnerability using a hardware-based memory protection mechanism. The proposed method involves modifying the page table entries to ensure that the memory mappings cannot be changed once they are marked as read-only. Overall, [2] provides valuable insights for researchers and practitioners interested in improving the security of the Linux kernel memory allocation system and highlights the need for more robust and effective mitigation techniques.

The paper [3] "Open source adaptive backup extensions for enterprise environment" presents a literature survey on open-source adaptive backup extensions for enterprise environments. The paper provides an overview of the existing open-source backup solutions and their features, advantages, and limitations. The authors in [3], highlights the importance of adaptive backup extensions that can provide dynamic and efficient data backup and recovery in large-scale enterprise environments. [3] also discusses various backup techniques and tools, such as differential backups, incremental backups, and deduplication, that can be used to optimize backup performance and reduce storage requirements. Overall, [3] provides a valuable review of the state-of-the-art open-source backup solutions and highlights the need for more research and development to improve the scalability and efficiency of backup solutions in enterprise environments.

The paper [4] "Research on improving fairness of Linux scheduler" presents a literature survey on the research conducted on improving the fairness of the Linux scheduler. [4] also discusses various approaches to improving the fairness of the Linux scheduler, such as process grouping, time-sharing, and priority adjustments. Overall, [4] provides a comprehensive review of the state-of-the-art research on the Linux scheduler and highlights the need for more effective and fair scheduling algorithms. [4] identifies the problems with the current Linux scheduler, such as unfairness and low performance, and proposes a new scheduling algorithm, called FAIR-QoS, that addresses these issues. FAIR-QoS aims to provide fairness among processes and improve system performance by dynamically adjusting the quantum and priority of processes based on their resource requirements and historical execution times.

The paper [5] "Implementation of Linux centralized user authentication and cloud storage in teaching" presents a literature survey on the implementation of Linux centralized user authentication and cloud storage in teaching. [5] provides an overview of the existing solutions for implementing centralized user authentication and cloud storage in educational environments, including LDAP, Samba, and ownCloud. The authors highlight the benefits of using Linux-based solutions for managing user accounts and files, such as increased security, scalability, and flexibility. [5] also discusses various use cases for implementing Linux centralized authentication and cloud storage in teaching, such as managing student accounts and assignments, facilitating collaboration, and providing remote access to course materials. Overall, [5] provides a valuable review of the state-of-the-art solutions for implementing Linux centralized authentication and cloud storage in educational environments and highlights the need for more research and development to improve their usability and effectiveness.

The paper [6] "Unleashing Full Potential of Ansible Framework: University Labs Administration" presents a literature survey on the use of Ansible framework for university labs administration. The paper provides an overview of the existing solutions for managing and automating IT infrastructure, such as configuration management tools and cloud-based platforms. The authors highlight the benefits of using Ansible for managing university labs, such as increased efficiency, consistency, and scalability. [6] also discusses various use cases for using Ansible in university labs administration, such as managing software installations, configuring network devices, and deploying virtual machines. The authors provide a detailed analysis of the challenges faced in implementing Ansible for university labs administration, including security concerns, lack of automation skills, and limited resources. Overall, [6] provides a valuable review of the state-of-the-art solutions for using Ansible framework in university labs administration and highlights the need for more research and development to address the challenges and improve the effectiveness of Ansible-based solutions.

The paper [7] "Performance Measurement of Processes and Threads Controlling, Tracking and Monitoring Based on Shared-Memory Parallel Processing Approach" presents a literature survey on performance measurement of processes and threads in shared-memory parallel processing environments. [7] provides an overview of the existing methods for monitoring and measuring the performance of processes and threads, including system monitoring tools, performance counters, and profiling techniques. The authors highlight the importance of measuring the performance of processes and threads in shared-memory parallel processing environments and provide a detailed analysis of the challenges and limitations associated with these environments. [7] also discusses various approaches to measuring performance, such as benchmarking, load testing, and stress testing, and their benefits and drawbacks. Overall, [7] provides a valuable review of the state-of-the-art methods for measuring the performance of processes and threads in shared-memory parallel processing environments and highlights the need for more research and development to improve their effectiveness and accuracy.

The paper [8] "Java Linux Administration Tool" presents a literature survey on the development of a Java-based tool for Linux system administration. [8] provides an overview of the existing tools and techniques used for Linux system administration, such as command-line utilities, graphical user interfaces, and web-based applications. The authors discuss about the benefits of using a Java-based tool for Linux system administration, such as platform independence, ease of use, and flexibility. [8] also discusses the design and implementation of the Java Linux Administration Tool (JLAT), including

its architecture, features, and capabilities. The authors provide a detailed analysis of the performance and usability of JLAT and compare it with other Linux administration tools. Overall, the paper provides a valuable review of the state-of-the-art tools and techniques for Linux system administration and highlights the potential benefits of using a Java-based tool like JLAT.

The paper [9] "High performance logging system for embedded UNIX and GNU/Linux applications" presents a literature survey on the development of a high-performance logging system for embedded UNIX and GNU/Linux applications.

[9] provides an overview of the existing logging systems for embedded systems, including syslog and custom logging solutions. The authors highlight the limitations of these logging systems, such as low performance, lack of flexibility, and limited storage capacity. [9] also discusses the design and implementation of the proposed logging system, including its architecture, features, and capabilities. The authors provide a detailed analysis of the performance and efficiency of the logging system and compare it with other logging solutions. Overall, the paper provides a valuable review of the state-of-the-art logging systems for embedded systems and highlights the potential benefits of using a high-performance logging system like the one proposed in the paper.

III. Objective

To automate the following Linus administrative tasks using bash shell scripts.

- Disk Space Status
- File Backup System
- Killing Stale Process
- User Account Creation
- Updating Clients Login Credentials

IV. Tools Used

Linux terminal commands are used to fetch the details of the system and user info. These details are then incorporated into the bash shell script. Cron is a time based job scheduler that runs scripts and commands periodically.

V. Proposed Architecture

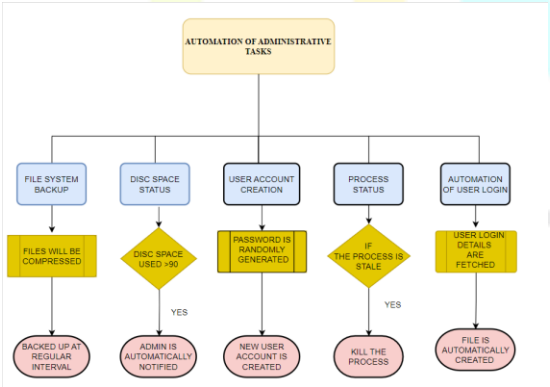

Fig. 1. Proposed architecture

VI. MODULES

A. Disc Space Status

In Linux, checking the disk space status is essential to ensure that the system is running optimally. When the user is busy working, they might not know whether the disc space is available or not. If the system run out of space, there is high chance that the system crashes. Checking disk space regularly can help prevent these issues by identifying when disk space is running low and allowing administrators to take action to free up space before it becomes a problem. The objective of this module is to notify the user if the disc space usage is greater than the threshold space.The threshold disc space is set in the shell script and the disc space of the system is retrieved using a command. Conditional statements are used to check whether the disc space used is greater than threshold space. If it is true,an alert will be triggered. The process is scheduled automatically using a tool like Cron, which is a time-based job scheduler.This allows system administrators to stay informed about the status of their system's disk space and take action before it becomes a problem.

Step 1: Start
Step 2: Use df command
Step 3: Parse the output and extract the value in number
Step 4: Store in discspace
Step 5: Set a threshold value
Step 6: IF discspace > threshold value
Print the alert message.
Step 7: Exit

B. File System Backup

Automating file system backups in Linux is crucial for data protection and disaster recovery.Backing up files regularly can prevent data loss in case of hardware failure and accidental deletion. A file system backup is a copy of all data on a file system, including files, folders, and permissions. Users who work on a daily basis need to backup their files or directories to another location in order to prevent losing them by accidently deleting it. Backing up the files daily is a tedious job as they might even miss doing it. This module gives a solution to this problem by automating the file backup by scheduling it daily. This automatically backs up all the files till date to another location, by converting the file type to zip. Regularly testing backups is important to ensure they are working correctly and can be used in the event of data loss.

Step 1: Start
Step 2: Change user to "root" using command "sudo -s"
Step 3: Check whether the backup file already exists
Step 4: IF 0
Create new file and file backup using backupfs.sh file
ELSE
echo " Backup already done"
Step 5: Exit

C. Killing the Stale Process

Stale processes are processes that have become unresponsive or stuck in an infinite loop, and are no longer performing

their intended function. These processes can consume valuable system resources and can impact the overall performance and stability of a Linux system. Killing the stale process in Linux refers to the process of terminating or stopping these unresponsive processes. This module focusses on killing such zombie processes automatically inorder to improve the system performance. The script uses the ps command to get a list of all running processes, sorts them by start time, and loops through each process. For each process, the script calculates the time since the process started and compares it to the threshold time. If the process is stale, the script uses the kill command to terminate the process. To automate this process using cron, the script can be saved to a file and then be scheduled to run at a specific interval using cron.

Step 1: Start
Step 2: Use ps command and find the sleeping and Zombie process
Step 3: Parse the output and extract the process id
Step 4: Store in killid
Step 5: Use grep command to filter the process
Step 6: Use kill command
Step 7: Echo the notification message
Step 8: Exit.

D. Client Login Credential

1. In this approach, the login credentials of all the clients is send to the system administrator at a particular time. This is achieved by using cron, a time-based job scheduler that is used in Unix-like operating systems to automate repetitive tasks. It allows users to specify commands or scripts to run at specific intervals, such as hourly, daily, weekly, or monthly. In this case, the cron job would be configured to run a script at a particular time based on the user convienence . A script is created that collects the login details of all the clients in approproiate format. The cron job is then configured on the server to run the script at a specific time using the crontab command. Then it is tested by running it manually using the script name or the full path to the script file. It is important to note that the script should be designed and tested carefully to ensure that it collects and formats the login details correctly and does not introduce any security vulnerabilities. In addition, the admin should ensure that the login details are transmitted securely and only to authorized personnels. Once the cron job is configured and tested, the login details of all clients will be sent to the admin at the specified time every day. This information is then stored in the admin's system as a file. The admin can then use this information to monitor the usage patterns of clients and detect any anomalies or issues that may arise.

Step 1 : Start
Step 2 : Use last command to fetch login history of users logged in.
Step 3 : Create a file to store the user details.
Step 4 : Use echo command to write the fetched user details to the file.
Step 5 : Store the file automatically in the admin's root folder using cron tab.
Step 6 : Exit

E. User Account Creation

Automation of user account creation refers to the process of creating user accounts automatically without requiring manual intervention. This process can be automated using various tools and techniques, such as scripting languages, configuration management tools, and identity management systems. Automated user account creation ensures that all user accounts are created in a consistent manner, without any human errors or variations. It saves time and effort and the accounts are created in secure and compliant manner. Manually creating user accounts especially in large enterprises where the number of user accounts is constantly growing can lead to delays and inefficiencies. This can also lead to inconsistencies in the settings and configurations of different user accounts. Instead of manually creating each user account, a script can be used to automate the process and create multiple accounts at once. Automating user account creation reduces the time and also ensures consistency in the accounts that are created. The steps involved in creating a user account are collecting user information, setting passwords, assigning roles and permissions, and setting up email accounts. This information is used to create a new account . Monitor the automated user account creation process to ensure that it continues to function as intended, and make updates or modifications as needed to accommodate changes in the organization39;s requirements.

Step 1 : Start
Step 2 : Get the username from the user.
Step 3 : If username already exists
Display the error message.
Step 4 : Use pwgen command to generate the password randomly.
Step 5 : Use useradd command to create new user account.
Step 6 : Log the creation of a new user account with date and time in a log file.
Step 7 : Use tee command to redirect to a log file.
Step 8 : Exit

Fig. 2. Process Flow

VII. JOB SCHEDULER

The cron daemon will automatically read the crontab file and schedule the specified process. For an example, the cron field is set as 00 17 * * * /backup.sh to create cron job that runs a backup.sh script to backup all the files at 5:00 pm on a daily basis.

Step 1 : Start
Step 2 : Use sudo apt - get install cron
Step 3 : Use sudo service cron status
Step 4 : IF cron not running
Use sudo service cron start
Step 5 : Use crontab –e to open crontab file
Step 6 : Schedule the .sh files
Step 7 : Set appropriate time and frequency for script execution
Step 8 : Exit

Fig. 3. Crontab editor

VIII. Conclusion

By automating tasks, we attempt to reduce the monotonous work of system administrator. Linux automation is a wide area covering multiple domains like cloud computing, IoT, networks etc. But, in this project we limit the scope and concentrate much in administrative tasks automation.

References

[1] M. R. Yaswinski, M. M. Chowdhury and M. Jochen, ”Linux Security: A Survey,” 2019 IEEE International Conference on Electro Informa- tion Technology (EIT), Brookings, SD, USA, 2019, pp. 357-362, doi: 10.1109/EIT.2019.8834112.

[2] D. Alam, M. Zaman, T. Farah, R. Rahman and M. S. Hosain, ”Study of the Dirty Copy on Write, a Linux Kernel memory al- location vulnerability,” 2017 International Conference on Consumer Electronics and Devices (ICCED), London, UK, 2017, pp. 40-45, doi: 10.1109/ICCED.2017.8019988.

[3] F. B. Manolache, Qingping Hou and O. Rusu, ”Open source adaptive backup extensions for enterprise environment,” 2013 RoEduNet Interna- tional Conference 12th Edition: Networking in Education and Research, Iasi, Romania, 2013, pp. 1-6, doi: 10.1109/RoEduNet.2013.6714206

[4] W. Wu, X. Yao, W. Feng and Y. Chen, ”Research on improving fairness of Linux scheduler,” 2013 IEEE International Conference on Information and Automation (ICIA), Yinchuan, China, 2013, pp. 409-414, doi: 10.1109/ICInfA.2013.6720333.

[5] Y. Chen and A. Zhu, ”Implementation of Linux centralized user authenti- cation and cloud storage in teaching,” 2014 International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, Japan, 2014, pp. 626-628, doi: 10.1109/InfoSEEE.2014.6948189.

[6] P. Masek, M. Stusek, J. Krejci, K. Zeman, J. Pokorny and M. Kud- lacek, ”Unleashing Full Potential of Ansible Framework: University Labs Administration,” 2018 22nd Conference of Open Innovations Association (FRUCT), Jyvaskyla, Finland, 2018, pp. 144-150, doi: 10.23919/FRUCT.2018.8468270

[7] K. H. Sharif, S. R. M. Zeebaree, L. M. Haji and R. R. Zebari, ”Performance Measurement of Processes and Threads Controlling, Tracking and Monitoring Based on Shared-Memory Parallel Processing Approach,” 2020 3rd International Conference on Engineering Technol- ogy and its Applications (IICETA), Najaf, Iraq, 2020, pp. 62-67, doi: 10.1109/IICETA50496.2020.93188000

[8] A. Bentiba, A. Mohmed and J. Zemerly, ”Java Linux Administration Tool,” 2006 IEEE GCC Conference (GCC), Manama, Bahrain, 2006, pp. 1-4, doi: 10.1109/IEEEGCC.2006.5686245.

[9] J. Jeong, ”High performance logging system for embedded UNIX and GNU/Linux applications,” 2013 IEEE 19th International Conference on Embedded and Real-Time Computing Systems and Applications, Taipei, Taiwan, 2013, pp. 310-319, doi: 10.1109/RTCSA.2013.6732232.