A SURVEY ON ELASTIC BLOCK STORAGE (EBS) IN AMAZON WEB SERVICES TO AUTOMATE THE EBS ROOT VOLUME EXTENSION

Abstract

In AWS cloud, EBS root volumes are a critical component of EC2 instances, providing persistent block-level storage for the operating system and applications. However, as workloads and data volumes grow, EBS root volumes can quickly become overloaded and require additional storage capacity. Extending an EBS root volume in AWS is a time-consuming process that involves several steps, including stopping the EC2 instance, creating a backup snapshot of the EBS root volume, modifying the volume size, and then restarting the instance and extending the file system.

To save time and reduce the risk of errors, automation can be used to streamline this process. There are several tools available in AWS that can be used to automate the extension of EBS root volumes, including AWS Systems Manager Automation, AWS Lambda, and AWS CloudFormation. AWS Systems Manager Automation provides a visual workflow editor that allows users to create a step-by-step process for extending EBS root volumes. This can be triggered manually or scheduled to run automatically based on specific criteria. AWS Lambda is a serverless computing service that can be used to create custom scripts that automate the process of extending EBS root volumes. This can be triggered by an event, such as a CloudWatch alarm that detects an overloaded EBS root volume. AWS CloudFormation is a service that allows users to define infrastructure as code, including the extension of EBS root volumes. This can be used to create a

Authors

N Sandeep Chaithanya
Department of Computer Science Engineering
VNRVJIET

Bandaru Siri
Department of Computer Science Engineering
VNRVJIET

Belide Prasanna Laxmi
Department of Computer Science Engineering
VNRVJIET

Bussa Sumukhi
Department of Computer Science Engineering
VNRVJIET

Nerellakunta Divya
Department of Computer Science Engineering
VNRVJIET

template that can be deployed across multiple instances, ensuring consistent and reliable storage expansion. By automating the process of extending EBS root volumes in AWS, organizations can save time, reduce the risk of errors, and ensure that their workloads and data have the storage capacity they need to function efficiently and effectively.

Keywords: AWS, EC2 instance, EBS root volume, Automation, CloudWatch Alarm, AWS Lambda, Serverless, Storage expansion

I. INTRODUCTION

Amazon Web Services (AWS) is a cloud computing platform that provides a wide range of services for computing, storage, and networking. AWS offers a scalable and flexible infrastructure to run various types of applications, ranging from simple web applications to large-scale enterprise applications. One of the core services of AWS is Elastic Block Store (EBS), which provides persistent block-level storage for Amazon EC2 instances. An EBS volume is a virtual hard disk that can be attached to an EC2 instance and used as the root volume, data volume, or for backup and disaster recovery purposes. The root volume in AWS is the primary storage device used by an EC2 instance to boot and run the operating system and applications. It contains the necessary files and configurations needed for the instance to function properly. The root volume can be an EBS volume, an instance store volume, or an Amazon Machine Image (AMI).

EBS root volumes provide several advantages over instance store volumes. EBS volumes are persistent, meaning that the data stored on them remains even after the instance is terminated. This allows for greater flexibility and reliability, as instances can be stopped and started at any time without losing data.

Additionally, EBS volumes can be easily backed up and restored, making them an ideal choice for disaster recovery scenarios. They also offer better performance and durability than instance store volumes, which are ephemeral and can only be used for temporary storage.

II. EXISTING SYSTEM

In AWS, EBS root volumes can be extended to increase the amount of storage available for an EC2 instance. Here's how to extend an EBS root volume in AWS:

1. **Stop the EC2 Instance:** Before extending the EBS root volume, it's important to stop the associated EC2 instance to avoid data loss or corruption.
2. **Create a Snapshot of the EBS Root Volume:** Next, create a snapshot of the EBS root volume as a backup. This ensures that the data is not lost if anything goes wrong during the extension process.
3. **Modify the EBS Root Volume:** After creating a backup, modify the EBS root volume to increase its size. This can be done using the AWS Management Console, AWS CLI, or SDKs. When modifying the volume, select the "Modify Volume" option and increase the size of the volume as needed.
4. **Start the EC2 Instance:** Once the EBS root volume has been extended, start the EC2 instance.
5. **Extend the File System:** After starting the instance, the file system must be extended to

III. LITERATURE SURVEY

1. Gabriella Laatikainena, Oleksiy Mazhelis a, Pasi Tyrvainen published a paper on Cost benefits of flexible hybrid cloud storage: Mitigating volume variation with shorter acquisition cycle in the year 2019.This study combines on-premise storage with cloud storage. By shortening the acquisition cycle, organizations can adapt to changing data

volumes quickly. As a result of this efficiency of system will increase and also it saves money, but there may be security issues and system become complex.

2. Shilpa Malhotra, Ms.Shakti Arora, Mr. Surjeet Singh published a paper on Amazon Backed File system with Enhanced Storage Feature in 2018 .In this papers it explains about In this hierarchical file system is preferred over the existing flat file system in AWS.The authenticated user can be able to store his data in the hierarchical file system in order to not access confidential data of other users. As the result performance will be increase and it is easy to scale across multiple regions and instances but it depends on Amazon s3 it costs more.

3. Purva Khot, Shraddha Padalkar, Vidhi Raghvani, Aayush Agarwal published a paper on Cloud Migration with Google Cloud Storage and AWS (S3 Service) in 2020. this research involves these steps:
   - **Assessment:** The first step involves assessing the current state of the on-premises storage infrastructure and identifying the data to be migrated.
   - **Planning:** The second step involves developing a migration plan, including selecting the appropriate cloud storage solution (in this case, either Google Cloud Storage or AWS S3) and determining the migration approach
   - **Pre-Migration:** The third step involves preparing the on-premises infrastructure and the cloud storage solution for migration
   - **Validation:** The fifth step involves validating the data after migration to ensure that it has been successfully migrated and is accessible in the cloud storage solution.

4. Hao Zhuang, Rameez Rahman, Pan Hui, Karl Aberer published a paper on Optimizing Information Leakage in MultiCloud Storage Services in 2018.
   They suggested the following steps:

   - **Threat Modeling:** The first step is to identify the potential threats to the multicloud storage service. This includes identifying the possible attackers and their motivations, as well as the vulnerabilities in the storage service
   - **Privacy Requirements Specification:** The second step is to specify the privacy requirements of the users of the storage service. This includes identifying the types of data being stored, the sensitivity of the data, and the desired level of privacy protection.
   - **Information Leakage Analysis:** The third step is to analyze the information leakage in the multicloud storage service. This includes identifying the possible sources of information leakage, such as data transfer between clouds, data replication, and data storage
   - **Optimization:** The fourth step is to optimize the information leakage in the multicloud storage service. This includes identifying the trade-offs between privacy protection and system performance, and choosing the best configuration of the storage service that satisfies the privacy requirements of the users while minimizing the information leakage
   - **Implementation:** The final step is to implement the optimized configuration of the multi-cloud storage service and evaluate its performance and effectiveness in protecting user privacy. It is used for organizations using multi-cloud storage services,

Data Privacy and OptimizationBut these won't be applicable to organizations that do not use cloud services.

5. Dr.Pierre, D. Boisrond published a paper on A Position Paper on Amazon Web Services (AWS) Simple Storage Service (S3) Buckets in 2021.They suggested few methods to increase security to s3 buckets one of them is to Ensure that AWS S3 buckets are not granting FULL_CONTROL access to authenticated users (i.e., signed AWS accounts or AWS IAM users) to prevent unauthorized access. The main advantage is Proper access controls can help to ensure the confidentiality, integrity, and availability of data stored in the S3 buckets, and can help to maintain compliance with regulatory requirements but there can be data breach and insider theft.

6. Valerio Persico published a paper on On the Network Performance of Amazon S3 Cloud-storage Service in 2016.This study tells us about CloudFront (CF) it is the global Content Delivery Network (CDN) service offered by Amazon and integrates with S3 in order to distribute contents to the end users with low latency and high data transfer speeds. Data is distributed to the users through the global network composed of the Amazon edge locations spread all over the world. By using Cloud Front users can have faster and responsive experience and it is to scale, but system become complex.

7. Xiangyao Yu, Matt Youill published a paper on Accelerating a DBMS Using S3 Computation in 2020.This paper is about Implemented a bare-bone row-based parallel DBMS testbed, called Push-downDB. PushdownDB represents a query plan as a directed acyclic graph of operators and executes in a pipelined fashion using multiple Python processes. A few performance optimizations are implemented, including disabling SSL as we expect analytics workloads are typically run in a secure environment and using the Pandas library to represent tuples as data frames. It can reduce data movements and leveraging of processing power but PushdownDB is currently only available for Amazon Redshift, so users who use other data warehouses are not able to take advantage of this service.

8. Patrick Cato published a paper on A Simple Approach to Optimize S3 Object Gateways for Massive Numbers of Small File Writes in 2022. He proposed 2 methods:
   - **CreateBucket method (handleContainerCreate):** will create one HBase table for object metadata, one HBase table for object data and a new volume on the distributed file system using the bucket name as the volume name and HBase table prefix
   - **PutObject method (handlePutBlob):** simple routing mechanism for small files using an if/else clause based on the length of the ByteArray was added. Data smaller than 500 KiB will be stored in the HBase table and files equal or greater than 500 KiB in the distributed file system.

     It has high security and highly scalable but it has high configuration complexity

9. Luan Teylo, Rafaela C. Brum, Luciana Arantes, Pierre Sens, Lucia Maria de A. Drummond published a paper on Developing checkpoint and recovery procedures with the storage services of Amazon Web Services. Identify the types of storage services offered by AWS that can be used for storing data and creating backups. Determine the requirements for checkpointing and recovery based on the application's characteristics and the level of data loss that can be tolerated. Develop a checkpointing strategy that

specifies the frequency of checkpoints, the data that needs to be saved, and the storage location.

10. C.S. Dzik, I.I. Piletski published a paper on Real time AWS resources monitoring and analytics. Data Collection: Collect data from various sources such as AWS documentation, whitepapers, and technical blogs. System Design: Based on the literature review and data collection, design a monitoring and analytics system for AWS resources. Implementation: Implement the designed system using appropriate tools and technologies. Evaluation: Evaluate the performance of the implemented system using various metrics such as response time, throughput, and resource utilization.

11. Ruben Matos, Jean Araujo, Vandi Alves, Paulo Maciel published a paper on Characterization of Software Aging Effects in Elastic Storage Mechanisms for private clouds. Data Collection: The researchers may have collected data by setting up a private cloud storage infrastructure, configuring elastic storage mechanisms, and running experiments to collect performance and resource utilization data. They may have used various workload scenarios to simulate realistic usage patterns. Analysis: The collected data would have been analyzed to identify patterns and trends that indicate the onset of software aging effects in the elastic storage mechanisms. Statistical methods may have been used to quantify the impact of aging on performance and resource utilization. Results: The results of the analysis would have been presented in the research paper along with their interpretation and implications for private cloud storage system design and management.

12. Aishwarya Anand published a paper on Managing Infrastructure in Amazon using EC2, CloudWatch, EBS, IAM and CloudFront Data Collection: Collect data from various sources such as AWS documentation, whitepapers, and technical blogs. System Design: Based on the literature review and data collection, design an infrastructure management system for Amazon using EC2, CloudWatch, EBS, IAM, and CloudFront. Implementation: Implement the designed system using appropriate tools and technologies. Evaluation: Evaluate the performance of the implemented system using various metrics such as response time, throughput, and resource utilization

13. Suma published a paper on Elastic Cloud Computing approach for Big Data Analytics
    - **Distributed File Systems:** This approach involves using distributed file systems, such as Hadoop Distributed File System (HDFS), to store and process large amounts of data across multiple computing nodes. Distributed file systems can provide fault tolerance and high availability for data processing.
    - **Multi-Cloud:** This approach involves using multiple cloud service providers to distribute the workload of the application across multiple cloud platforms. Multi-cloud can provide better fault tolerance and reduce vendor lock-in risks.

14. Er. Krishan Kumar, Shipra published a paper on Detailed Study of Cloud Storage Services on Aws Cloud (Amazon Web Services). Read AWS documentation - AWS provides detailed documentation on all their cloud storage services, including Amazon S3, Amazon EBS, Amazon EFS, Amazon Glacier, Amazon FSx, and Amazon Storage Gateway. Experiment with AWS cloud storage services 2. AWS provides a free tier that allows users to experiment with their cloud storage services without incurring any

charges. This provides an opportunity to test the service, understand its features, and evaluate its suitability for different use cases.

15. Priyadarshini P, K. T. Veeramanju published a paper on A Systematic Review of Cloud Storage Services- A Case Study on Amazon Web Services.
    - **Inclusion/Exclusion Criteria:** Once studies are identified, they are screened based on pre-defined inclusion and exclusion criteria. These criteria typically relate to the research question and may include factors such as publication date, study design, and type of cloud storage service.
    - **Quality Assessment:** The quality of the included studies is assessed using a standardized tool. This may involve assessing factors such as the validity and reliability of the study methods.
    - **Synthesis:** The data extracted from the included studies is synthesized and analyzed to answer the research question. This may involve statistical analysis or a narrative synthesis of the findings.

IV. EXISTING SYSTEM

When the monitoring reveals that there is insufficient free space in the EBS storage, manual intervention becomes necessary. Extending the storage capacity requires following a series of steps to ensure the process is carried out correctly. These steps typically involve identifying the target EBS volume, calculating the required extension capacity, executing the necessary commands or operations, and validating the success of the extension. However, this manual process can be time-consuming and may introduce the risk of human error.

To address these challenges and save time, there is a need to automate the process of extending EBS storage. By implementing automation techniques, such as scripting or infrastructure-as-code tools, it becomes possible to streamline and expedite the steps involved in extending the storage. Automation eliminates the need for manual intervention and reduces the chances of errors, making the process more efficient and reliable. By automating the extension of EBS storage, AWS administrators and operators can save valuable time and ensure that the storage capacity is extended seamlessly when needed.

V. PROPOSED SYSTEM

To launch an EC2 instance with an attached EBS volume, you can use the Amazon EC2 service. This allows you to create virtual servers in the cloud. During the EC2 instance creation process, you can specify the desired EBS volume and attach it to the instance. The EBS volume serves as persistent storage for your EC2 instance.

Once the EC2 instance is up and running, you can monitor its performance and resource utilization using Amazon CloudWatch. CloudWatch provides various metrics, including EBS storage metrics, to help you keep track of your EC2 instance's health and performance. You can create a CloudWatch alarm to watch these metrics specifically related to EBS storage.

With the CloudWatch alarm in place, you can configure it to trigger an action when specific conditions are met. In this case, you can use AWS EventBridge, a serverless event bus service, to respond to the alarm triggering. EventBridge acts as a bridge between various AWS services and enables event-driven architectures. You can set up an EventBridge rule to detect when the required condition, such as low available storage on the EBS volume, is met.

When the EventBridge rule detects the specified condition from CloudWatch, it can trigger an AWS Lambda function. AWS Lambda is a serverless compute service that allows you to run code without provisioning or managing servers. You can write a Lambda function that is designed to handle the triggered event and increase or extend the EBS root volume storage attached to the particular EC2 instance.

The Lambda function can utilize the AWS SDKs and APIs to interact with the EC2 service and perform the necessary operations. It can dynamically adjust the EBS root volume storage based on your predefined logic or requirements. This automation helps ensure that your EC2 instance's storage capacity is appropriately managed and extended when needed, without manual intervention.

By combining EC2, EBS volumes, CloudWatch, EventBridge, and Lambda, you can create a scalable and automated solution that monitors and adjusts the storage of your EC2 instances as per your defined conditions and business needs.

VI. METHODOLOGY

In the described system architecture, the automation of EBS root volume extension in AWS is achieved through the following steps: The system starts with an alarm that monitors the state of the EBS root volume. When the alarm state changes, indicating that the root volume requires extension, it triggers a Step Function. The Step Function is a serverless workflow orchestration service provided by AWS. It receives the trigger from the alarm and executes a sequence of Lambda functions in a defined order. The first Lambda function executed by the Step Function is called GETEBSID. This function retrieves the unique identifier (ID) of the EBS root volume that needs to be extended. It may use AWS SDKs or APIs to interact with the AWS environment and fetch the required information.

Once the GETEBSID function completes its execution, the Step Function moves on to the next Lambda function, GROWEBS. This function performs the actual extension of the EBS root volume. It utilizes the EBS ID obtained from the previous step to interact with AWS services and increase the size of the volume. After, the GROWEBS function finishes extending the EBS root volume, the Step Function proceeds to the final Lambda function in the sequence, GROWFS. This function is responsible for resizing the file system within the extended volume, ensuring that the operating system recognizes and utilizes the additional storage space. Once the GROWFS function completes its execution, the Step Function concludes, and the system can observe that the alarm state changes to a safe state. This indicates that the EBS root volume has been successfully extended, and the system is now utilizing the increased storage capacity.

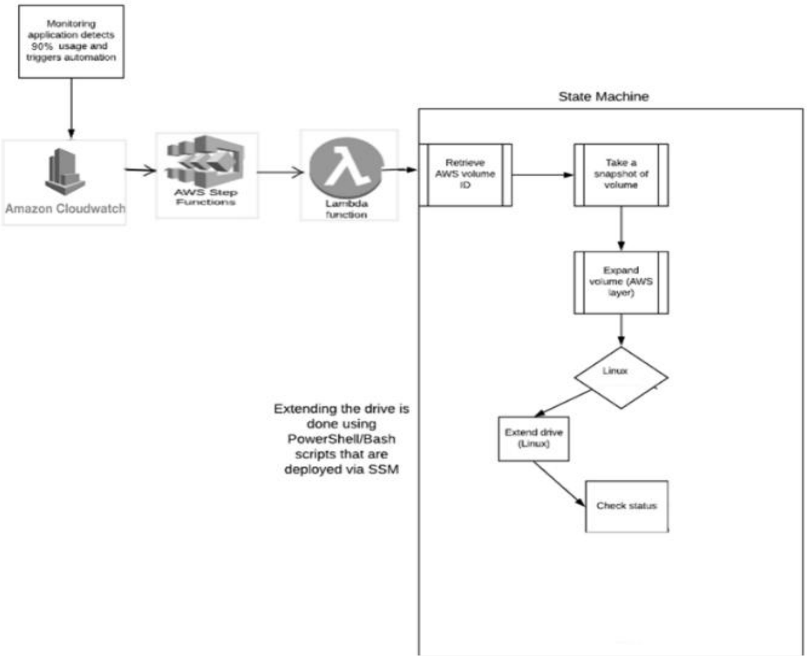
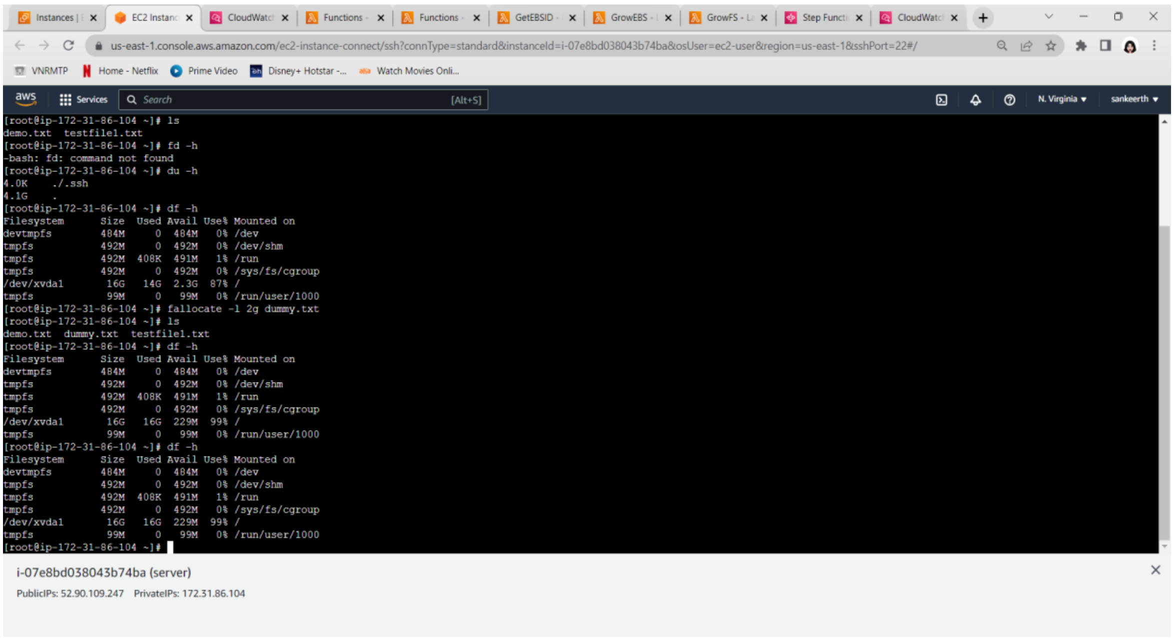

Monitoring application detects 90% usage and triggers automation
Amazon Cloudwatch
AWS Step Functions
Lambda function
State Machine
Retrieve AWS volume ID
Take a snapshot of volume
Expand volume (AWS layer)
Linux
Extend drive (Linux)
Check status
Extending the drive is done using PowerShell/Bash scripts that are deployed via SSM

Figure 1

VII. RESULTS



```
[root@ip-172-31-86-104 ~]# ls
demo.txt  testfile1.txt
[root@ip-172-31-86-104 ~]# fd -h
-bash: fd: command not found
[root@ip-172-31-86-104 ~]# du -h
4.0K    ./.ssh
4.1G    .
[root@ip-172-31-86-104 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        484M     0  484M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  408K  491M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1       16G   14G  2.3G  87% /
tmpfs            99M     0   99M   0% /run/user/1000
[root@ip-172-31-86-104 ~]# fallocate -l 2g dummy.txt
[root@ip-172-31-86-104 ~]# ls
demo.txt  dummy.txt  testfile1.txt
[root@ip-172-31-86-104 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        484M     0  484M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  408K  491M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1       16G   16G  229M  99% /
tmpfs            99M     0   99M   0% /run/user/1000
[root@ip-172-31-86-104 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        484M     0  484M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  408K  491M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1       16G   16G  229M  99% /
tmpfs            99M     0   99M   0% /run/user/1000
[root@ip-172-31-86-104 ~]#
```

i-07e8bd038043b74ba (server)

PublicIPs: 52.90.109.247 PrivateIPs: 172.31.86.104

Figure 2

The volume of the EBS has increased automatically. we can confirm it by using the command “**df -h**” in terminal or by checking the dashboard of the EC2 instance in AWS console.

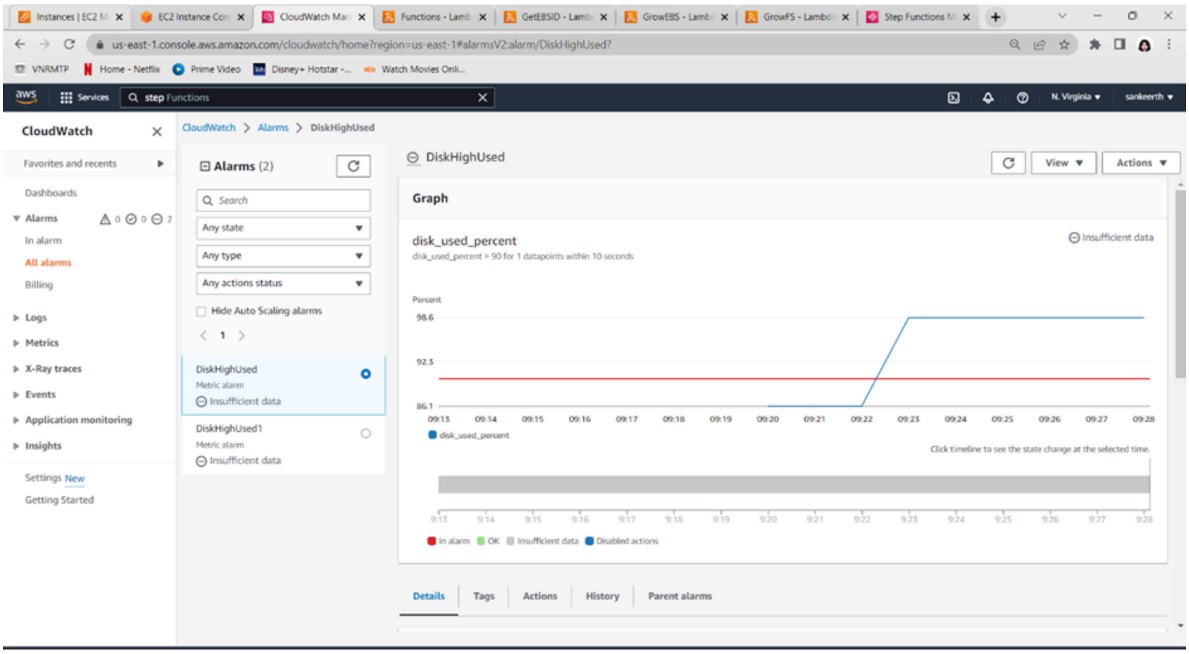


Figure 3

VIII. CONCLUSION

In conclusion, automating the process of extending EBS (Elastic Block Storage) root volumes in AWS cloud can greatly benefit users by saving time and streamlining operations. The current manual process of extending EBS storage can be time-consuming and may require significant effort. By implementing automation, the steps involved in extending EBS volumes can be automated, reducing human intervention and potential errors. This automation can help ensure that when the root volume is about to get overloaded, the necessary storage extension is carried out promptly and efficiently. Overall, automating this process will enhance operational efficiency, improve productivity, and enable users to effectively manage their EBS volumes in AWS cloud environments.

REFERENCES

[1] Cost benefits of flexible hybrid cloud storage: Mitigating volume variation with shorter acquisition cycle. Author: Gabriella Laatikainena, Oleksiy Mazhelis a, Pasi Tyrvainen. Published by: Elsevier
[2] Amazon Backed File system with Enhanced Storage Feature. Authors: Shilpa Malhotra, Ms.Shakti Arora, Mr. Surjeet Singh. Published By: IJCRT
[3] Cloud Migration with Google Cloud Storage and AWS (S3 Service) Authors: Purva Khot, Shraddha Padalkar, Vidhi Raghvani , Aayush Agarwal. Published By: IEEE
[4] Optimizing Information Leakage in Multi-Cloud Storage Services. Authors: Hao Zhuang, Rameez Rahman, Pan Hui, Karl Aberer. Published By: IEEE
[5] A Position Paper on Amazon Web Services (AWS) Simple Storage Service (S3) Buckets. Author: Dr.Pierre D. Boisrond. Published By: IEEE
[6] On the Network Performance of Amazon S3 Cloud-storage Service. Author: Valerio Persico. Published By: IEEE
[7] Accelerating a DBMS Using S3 Computation. Published by: IEEE
[8] A Simple Approach to Optimize S3 Object Gateways for Massive Numbers of Small File Writes. Published By: IEEE

[9] Developing checkpoint and recovery procedures with the storage services of Amazon Web Services. Authors: Luan Teylo, Rafaela C. Brum, Luciana Arantes, Pierre Sens, Lucia Maria de A. Drummond. Published By: IEEE

[10] Characterization of Software Aging Effects in Elastic Storage Mechanisms for private clouds. Authors: Ruben Matos, Jean Araujo, Vandi Alves, Paulo Maciel. Published By: IEEE

[11] Real time AWS resources monitoring and analytics. Authors: C.S.Dzik, I.I.Piletski. Published By: IEEE

[12] Managing Infrastructure in Amazon using EC2, CloudWatch, EBS, IAM and CloudFront Author: Aishwarya Anand. Published By: IEEE

[13] ELASTIC CLOUD COMPUTING APPROACH FOR BIG DATA ANALYTICS. Author: Suma. Publication title: Journal Of Ubiquitous Computing and Communication Technologies

[14] Detailed Study of Cloud Storage Services on Aws Cloud (Amazon Web Services). Authors: Er. Krishan Kumar , Shipra. Publication title: International Journal Of Scientific Research In Computer Science, Engineering And Information Technology

[15] A Systematic Review of Cloud Storage Services- A Case Study on Amazon Web Services. Author: Priyadarshini P. , K. T. Veeramanju. Publication title: Cern European Organization For Nuclear Research - Zenodo