International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Automated Notification for Improper Use of AWS Resources

Pragya Nayak, Jenila Livingston L. M.

Abstract*: A proper management of resources is very important when they are on public cloud to ensure security, cost optimization etc. Monitoring resources and sending notification to admin or authorized person for all monitored activities helps the corporate institute/organization who is using public cloud like AWS to manage the resources properly, as AWS in an organization is used by different users though they have roles associated with them but they can do any mistake which costs for an organization. Automated notifications can help on this issue when resources are monitored and notified to admin/authenticated person and they can take appropriate actions. This paper focuses on how to make an environment and hardware setup of instance in AWS using AMIs, user data and Terraform, securing and managing S3 from unwanted objects and automated deletion of unwanted resources. It is also important for an organization to verify that developers, tester etc. are working according to their provided environment standard so that they can guarantee that there is not be any problem after or during project development or delivery so, in context of that AMIs can be created and EC2 monitored for AMI.

***Index Terms*: AWS - Amazon Web Service, resource monitoring, SES - Simple Email Service, SNS - Simple Notification Service, S3 - Simple Storage Service, EC2 - Elastic Compute Cloud, Jenkins, Terraform**

I. INTRODUCTION

The emergence of cloud computing and public cloud comes with various advantages and disadvantages the major one is security. Public cloud providers like Amazon Web Service offers different cloud services but when it is used in a corporate organization some measures should be taken care like cost and security. The improper or invalid use of AWS Resources indicates that the resources being used by user in an AWS Account is not matching the standard provided by the organization. It is important to follow the standard as discussed in this paper for security, cost and time. Currently, there are around 119 services provided by AWS and 16 regions, where each region has minimum of 2 availability zones and maximum 6 availability zones (a, b, c, d, e, f) and in AWS Organizations there are multiple accounts or group of accounts that can be controlled centrally with help of SCPs (Service Control Policies). AWS Organization provides a simple way of payment that is single payment method for multiple accounts of AWS. There is no additional charge for AWS Organization. So, it is a best practice to use AWS Organization as there are different groups, teams and works within an institute/organization which can be separated with AWS Organization. The most used AWS resources are EC2 and S3. Amazon EC2 (Elastic Compute Cloud) is one of the most popular virtual server on public cloud. It is using all compute resources of AWS infrastructure for example storage, CPU etc. and the AWS user need to pay for what they are using and that is the beauty of cloud that you have to pay only for what you are using that is called pay-per-use feature of cloud. AWS provides its resources like EC2 on rent and it has its billing criteria according to which it bills for its resources used. EC2 is a virtual machine that is hosted on AWS infrastructure and is made using AMI (Amazon Machine Image) which is consist of launch permissions, block device mapping and template (example application servers, operating system, application etc.), security groups which is like firewall for EC2 and give the permission regarding inbound and outbound traffic, etc. In this paper a custom AMI is created using Terraform so that it can contain all the necessary configurations for the virtual server. The configurations or installations can be done using user data that we create while launching of EC2 instance. This helps corporate organization in faster delivery of application, ensure security and reducing the time because the AMI made up of user data contains all security related software and dependent software to create an application and this also helps developer to focus only on the coding part i.e. the development part and not the complete setup for the development of application. Unlike EC2, AWS Lambda is known as 'Serverless Platform' that is AWS is take care of server and we need not to manage any server related stuff. Also it provide maximum of 15 minute execution time of maximum of 3008 MB RAM which can be configured. AWS charges lambda on basis of number of invocation made. AWS S3 or Simple Storage Service is another very popular and important service and it is a storage service provided by AWS where one can store different kind of files, host an application on S3 and it has various features like encryption, versioning etc. S3 consist of bucket and its name should be unique among all the buckets of AWS accounts present and these buckets have objects in them, objects can be different form of files. There can be any number of objects in S3 but each object can have maximum of 5 terabyte size.

Revised Manuscript Received on 30 July 2019.

***** Correspondence Author

Pragya Nayak*, M. Tech, School of Computing Science and Engineering, Vellore Institute of Technology University, Chennai, Tamil Nadu, India -600127.

Jenila Livingston L.M., Associate Professor, School of Computing Science and Engineering, Vellore Institute of Technology University, Chennai, Tamil Nadu, India -600127

Similar to EC2, S3 also needs to be maintained and secured for example if anyone made a S3 bucket as public or done any wrong permission setting in its ACL then anyone can access it and confidential information can be leak. To ensure its security a proper maintenance required which includes automated notification so that whenever anyone try to make changes in bucket or its object like make it public or unencrypted then it should be monitored and an notification should be send to an authorized person so that they can take a proper action or it should be deleted or configuration should be changed in proper time before anyone misuse it. Notifications can be sending through AWS SNS (Simple Notification Service) and AWS SES (Simple Email Service). SNS uses publish subscribe model and there can be many subscriber for a topic which can get notification. These subscriber also includes AWS Services for example a lambda can be one subscriber and can be invoked as a publisher publishes in a topic, similarly SQS can be a subscriber, etc. So SNS is helpful when we want any immediate action to take place whenever publisher publish in a topic for example if anyone put any exe file it should be deleted or notified, in this case notification can be send directly but if exe file should be deleted directly then they should call a lambda which includes a code to delete that object or when there is an E Commerce kind of application and after the placement of an order by user, details should be received by shipping, item count and other related subscriber in that case SNS is a good option to complete the requirement. But if we want to send only email as a notification and not want to notify directly to any AWS resource or service then SES is a good option as it sends email as a notification to recipients where the sender and receivers should be first registered to SES. SES currently not available to all the regions of AWS and there are many resources which are present for all of regions of AWS. Terraform is an Open Source tool or project provided by Hashicorp, it helps to write infrastructure as a code (IaC) and create, modify and delete cloud resources. AWS infrastructure can be created with Terrform and also with AWS CloudFormation but AWS CloudFormation scope is limited to AWS Resources only whereas Terraform scope includes AWS Resources and also other third party cloud providers. Terrform is also different from configurations tools like chef and puppet as they helps to write configurations of infrastructure or software as a code or to code for configurations of system whereas IaC is used to create a system. The traditional approach in software development life cycle is to build an application first then testing then delivery and when any error occur or there is any requirement in code changes then again we have to repeat the cycle, later agile methodology came which breaks product into small pieces and these pieces integrate in final testing. It fills the gap between customer and its software requirement with developer and tester as agile focuses mainly on constant changes. But in Agile there is still gap between development team and operation team which can be filled with DevOps, unlike Agile DevOps focuses mainly on constant testing of product and its delivery. Jenkins is one of the tools of DevOps and is made up on java language which makes it platform independent. Jenkins helps in continuous integration and automated testing of developers build.

II. LITERATURE SURVEY

[1] Pedro Álvarez et al. has described that a migration of service from on-premise to cloud is beneficial in term of many aspects one of them is cost as if different category of applications to deployed on private infrastructure i.e. on premise need to have different infrastructure requirements like processors, storage devices and many physical devices and then they require maintenance like cooling, hardware failure protection etc. This increases the expenses. There are different public cloud options for this issue which provides Infrastructure as a service and cost is based on the usage of these infrastructures. When AWS EC2 (Elastic Compute Cloud) is used to deploy and execute an application, its pricing depends on various parameters like instance types, processor etc. The pricing policy for Amazon S3 depends on the count of read/write request, data stored and transferred into and from S3. Their proposed method has considered cost factors like data management cost i.e. cost for I/O operation and data storage, and framework component cost. [2] It can be easy to use resources of public cloud according to application requirement, Huankai Chen et al. states that management of these resources is an NP-complete problem because resource management complexity increases when application is deployed on cloud. The cloud complexity can delineate in different ways like the different cloud components for resource management are directly/indirectly connected and interact with each other and the failure of one can cause the complete resource management failure. Key characteristics for occurrence of resource management complexity includes the number of resources needs to manage by the resource management system, when a resource management system is managing large number of resources it increases the complexity and this also include that the change in any resource can change the level of complexity this is called *Numerousness*. *Interdependency* is another key characteristic which means direct/indirect connection and relation in cloud resource and this can increase the level of complexity. [3] Xiaolong Liu et al. discuss about AWS CloudWatch (CW) tools, AWS CloudWatch is cloud resource management frameworks of AWS and it is a monitoring tool used by AWS which uses metrics to monitor the AWS resources. This monitoring allows for doing some triggers in specific conditions. There are different metrics available in CW with respect to the AWS resources, metrics included in AWS CW Disk read IN/OUT, CPU Utilization, Network IN/OUT etc., and also some other custom metrics. These metrics are applied to AWS resources and monitor them and when the metrics reach the specific threshold an action will take place like if it is an ASG (Auto Scaling Group) and ELB (Elastic Load Balancer) then new instance will be created because of CW threshold and traffic will be transferred to that instance. Similarly, CW is used with other AWS resources too. [4] Public cloud provides scalable and elastic services to the external customers/users via internet.

The services of public cloud used by organization sometimes need to be in some specific standard or pattern from user side this might because of theirs organization standard and that standard need to be follow because of organization's security concern for example they always want their stored data on public cloud to be encrypted. Daniel Sun et al. has focuses on Amazon Web Service public cloud. System administrator, developer or any authorized user can access metrics by using public cloud facility like AWS CloudWatch. AWS CloudWatch provides the facility to monitor AWS cloud resources, and it also provides facility for customer defined metrics. Though AWS CloudWatch provides monitoring facility using metrics but it is not possible to monitor and detect anomalies or improper use of AWS cloud resources just by using AWS monitoring tool CloudWatch. [5] When data is stored on cloud or when data is migrated from on-premise to cloud, it is beneficial with respect to owning a storage devices on private infrastructure because of maintenance complexity, overhead and cost of the storage devices but when data is stored in cloud then security becomes one of the major concern according to Jan Stanek et al. the solution of this security concern is encryption of the data stored over the cloud. As corporate and private users are increasing the use of cloud storages, breaching of data is also increasing specially for sensitive and popular data therefore, data is needed to be secure and hence encrypted. [6] Pieter-Jan Maenhaut et al. has described about the importance of resource management in cloud computing. Cloud computing provides on demand service to the end user and end users save the cost as they do not have to own and pay for the infrastructure but only for what they are using. Though to use Public clouds, its services and resources, customer needs to pay for what they are using but if they are not utilizing and using the resources in proper way then they have to pay for unnecessary things which they are not even using and all this is because of improper management of resources and therefore efficient and proper management of resources in public cloud is very important and it results in cost reduction. When resources are created and allocated they should follow some strategies and designs and for that they need to be validated. [7] Serverless infrastructure AWS lambda provides function as a service model. It is serverless because unlike AWS EC2 instance end user does not configure and create it but end user only does coding on it as Lambda is compatible with different programming languages. Since February 2019 the programming languages and the runtimes supported by lambda are Node.js 6.10, Node.js 8.10, Python 2.7, Python 3.6, Python 3.7, Ruby 2.5, Java 8, Go 1.x, .Net Core 1.0, .Net Core 2.0 and .Net Core 2.1, also there is custom runtime which allows to use any other runtime in AWS Lambda. The Lambda server is managed by AWS. Maciej Malawski et al. have used serverless approach of AWS Lambda to process the background tasks. [8] Similar to data security over the cloud, accessibility of data over the cloud is also a major concern i.e. who can access the data that is stored on cloud. The three main entities of cloud computing are *cloud service provider* and he will manage the cloud services, *data owner* will store the data on cloud and *user* will send request to access the data from cloud stored by data owner or to access any other kind of cloud services. Access control is a mechanism which can be defined as a accessing authority for any kind/type of data and files from different cloud services and cloud resources. Suyel Namasudra et al. proposed a method to access the data on cloud efficiently and this method consist of operation like user authorization i.e. user who is allowed to access data stored on cloud must be registered with the respective cloud service provider and later user authenticity will be check from cloud service provider and allow or disallow the accessibility of stored data. While storing the data, data owner generates the secret key and encrypt the data with the generated secret key. The stored data can be inserted, deleted and searched. [9] Nguyen Cong Luong et al. have presented a review of pricing for resource management. Resource management should have robust and adaptive design for cost reduction, provisioning flexibility of cloud resource and this design helps to solve issues like resource allocation, request allocation etc. Because of user satisfaction, profit, resource utilization etc. pricing model has taken into consideration [10] Through IaaS (Infrastructure as a Service) of cloud computing cloud customers can get highly flexible and high availability of resources, tenants can leverage this IaaS cloud service and deploy there application and work on them with many advantages of IaaS but as the infrastructure is in remote location owned by some third party security becomes a concern and user avoid to use cloud services. Nicolae Paladi et al. describe a framework for IaaS operation security and data security. The framework consists of key management, encryption etc and also other additional measures to ensure the security and enhancement of cloud infrastructure.

III. PROPOSED METHOD

This paper focuses on security and management of AWS EC2 and AWS S3 and for that a standard or a pattern is defined for these resources that should be followed by them and if EC2 and S3 are not following the standard then they are referred as resources with improper data usage. Other part of management of AWS resources include automated deletion of AWS resources which are not valid according to the standard. So, whenever someone creates invalid or improper resource then an automated SES/SNS notification is sent to admin with the detail of resource. The resources which are following the standard are saved from the automated deletion and their IDs are saved in a S3 bucket. The S3 bucket includes all the valid information of resources in a configuration file refer to as exclusion list and for different resources there are different configuration files example there is individual file for EC2, ASG (Auto Scaling Group), S3 etc. where the files of exclusion list are excluded from automated notification and deletion. Only master or admin that is the authorized persons are the recipient and have access to store and delete file from exclusion list. The programming part that is doing the automation is done in python language and AWS Lambda is used as a platform for this automation. Now, to make this programming as automated lambda needs to invoke repeatedly in some interval and this can be achieved through AWS CloudWatch or Jenkins.

We can create an event in AWS CloudWatch with cron expression which invokes Lambda functions according to the created event or we can use Jenkins that is known for continuous integration and can be configured and scheduled to invoke AWS Lambda in specified interval. We have configured pipeline of Jenkins to invoke the Lambda function and install and configure AWS CLI using shell script and then call AWS Lambda and set schedule time for example as each Friday 10 AM. It ensures continuous integration and whenever any changes made in lambda it is directly reflect in AWS account. Appropriate IAM roles are added for Lambda to access AWS resources via Lambda.

Fig 1 shows the complete architecture of proposed method. The proposed method is divided into four modules:

- Module1: EC2 Custom Image (AMI) Creation
- Module 2: Python code for automated verification of EC2 Instances
- Module 3: Python code for automated SNS/SES notification regarding S3
- Module 4 (a): Alert notification for automated resource deletion.
- Module 4 (b): Automated AWS resource deletion

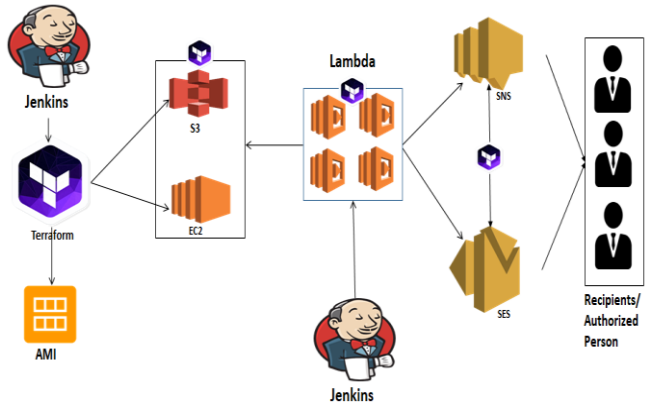

Fig 1: Architecture for Automated Management of AWS Resources

MODULE 1: EC2 CUSTOM IMAGE (AMI) CREATION

According to our proposed method, the standard that should be followed by AWS EC2 instance is that it should be spin up with the custom image. The custom image contains all the required configurations for the instance i.e. the configurations that are required to complete any project. Fig 2 shows the flow for Module 1 and the steps are as follows:

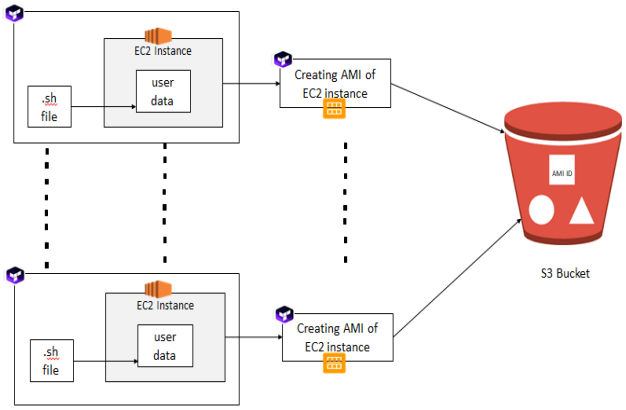

Fig 2: Workflow for EC2 Custom Image Creation

Step 1: Creating EC2 instance with user data:

- Creating EC2 with Terraform: It includes AMI, subnet, VPC, security group, Amazon CloudWatch metric and other relevent details to create an EC2 where the CloudWatch metric is optional

- Adding user data with uploading .sh file: User data conatins the sh file or the configuarion commands and it'll be added in EC2. The sh file contains the configuarations for instance and EC2 is created with the configurations available in user data.

Step 2: Creating custom AMI:

- Terraform code to create AMI using EC2 instance created in step 1

Step 3: Saving/Storing the AMI:

- Store AMI IDs in S3 bucket, bucket contains all the valid AMIs that should be used within an organizations and AMI IDs are uploaded in bucket by some authorized person.

MODULE 2: PYTHON CODE FOR AUTOMATED VERIFICATION OF EC2 INSTANCES

Whenever anyone want to spin an EC2 instance then the instance should follow some parameters to create the instance like instance should use only the custom image, encrypted AWS EBS (Elastic Block Storage) volume, specific instance type because of the features and pricing of instance types, defined security group because improper use security group can affect the instance privacy and security and EC2 should be spin up in specific region because of latency in regions, availability of AWS services in a region and cost variations of AWS regions. Admin knows the owner of EC2 instance which is possible by AWS CloudTrail

Fig 3 represents the workflow of module 2. The algorithm for automated verification of EC2 instances is as follows:

- Fetch all running or stopped instances present in AWS account for all the regions

- Comparing and verifying the instances with valid EC2 information stored in S3 bucket (Valid instance information can also be included in array variable that can be compared with fetched instance information). The code sends notification to authorized person if:

- AMI of instance is not present in S3 bucket containing valid AMIs IDs
- Region of instance that should not be used by users to create instance
- EBS Volume is not encrypted
- Valid instance type is not used to create EC2
- Subnet mismatches with required subnet
- Security group mismatches with required security group

- After receiving the notification, authorized person can asks the owner of the instance to make changes according to standard or changes are not required then admin adds the instance ID in S3 bucket otherwise the instance will be deleted (because of Module 4 (b): automated AWS resource deletion)

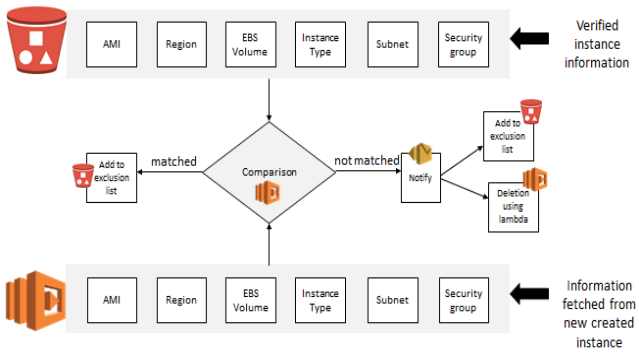

Fig 3: Workflow for automated verification of AWS EC2

MODULE 3: PYTHON CODE FOR AUTOMATED SNS/SES NOTIFICATION REGARDING S3

When the file(s) uploaded in S3 bucket do not follow the standard then here those files are defined as invalid files or improper files. A notification is send with each invalid file upload in S3 bucket. Fig 4 shows when a notification send to admin and later a deletion take place.

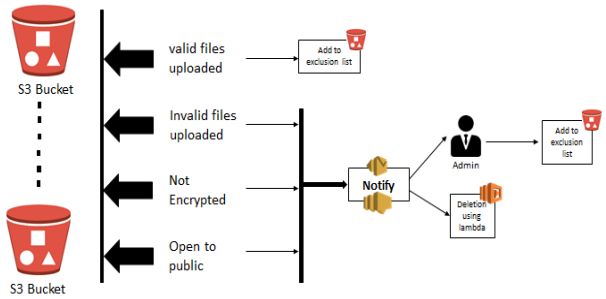

Fig 4: S3 SNS/SES Notification and Deletion for files that are not following the standard

1. AWS Lambda Function Sending Notification for Invalid/Improper Files Uploaded:

SNS notification is send to admin if someone upload a file that does not match the standard for example exe files are not allowed by a institute because it may be some monitoring software that can affect the privacy of organization/institute. To achieve this automated lambda code has been made using python:

- Using python library boto3 create publisher (S3) and subscriber (email) for SNS. SNS send immediate notification to subscriber (authorizer) as a mail.

- Add suffix value as a condition therefore, whenever a file upload with that suffix then SNS sends notification. Python code provide automation for SNS to notify invalid upload of object in S3 and notify when someone upload file with given suffix in s3 bucket for example exe file, tar file, zip file, etc. (Addition of prefix and suffix condition can also help to follow some standard naming convention for S3 objects)

2. AWS Lambda Function Sending Notification for Unencrypted Files in S3 Bucket:

- Lambda takes each of the S3 key information of S3 Bucket and check if key's server_side_encryption value is 'None' or if it is not using encryption then it sends notification using SES to authorized person.

3. AWS Lambda Function Sending Notification for Public Files in S3 Bucket:

- If Grantee's URI of object ACL is set as http://acs.amazonaws.com/groups/global/AllUsers it means object is accessible to public and notification is send using SES to authorized person

After notification, if objects of bucket are required to not follow the standard and it does not harm the organization then admin adds it to exclusion list.

MODULE 4 (A): ALERT NOTIFICATION FOR AUTOMATED AWS RESOURCE DELETION

There can be different resources created by users in an AWS account and they might be using invalid data therefore need to be deleted but before deletion an alert is sent to an authorized person who can make changes in the exclusion list and prevent the resource from automated deletion by adding resource in exclusion list or remove the excluded resource from being deleted and it will be deleted. Fig 5 explains how lambda is sending notification for deleting and excluding resource of AWS where deleting resources are the resources which are not in exclusion list and deleted whereas excluded resources are the resources which are in exclusion list and they are not deleted.

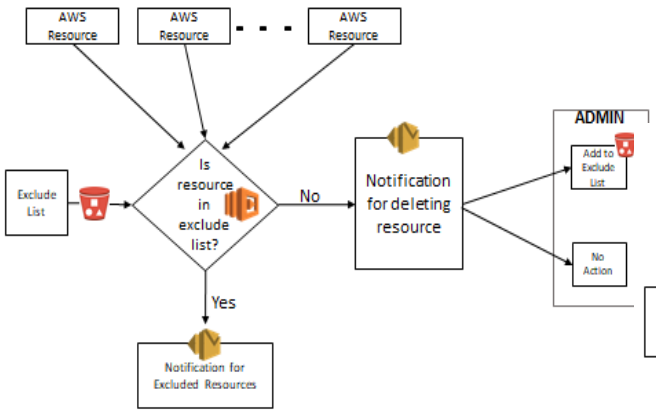

Fig 5: Alert Notification for Automated AWS Resource Deletion

The algorithm of alert notification for automated AWS resource deletion:

- Lambda compares AWS Resource with its respective configuration file containing its ID in S3 bucket exclude list.
- If AWS Resource is in exclude list, it sends notification for excluded resource otherwise, it sends notification as deleted resources
- Admin on receiving the notification can add or remove AWS resource ID from exclusion list

MODULE 4 (B): AUTOMATED AWS RESOURCE DELETION

The resources that are deleted from automated deletion code with their order are as follows:

- Auto Scaling Groups (ASG)
- EC2 Instances
- EBS Volumes
- AMIs
- EBS Snapshots
- Load Balancers(Classic,Application and Network Loadblancers)
- Target Groups
- Rds Instances
- Rds Snapshots
- DynamoDB Tables
- Elasticbeanstalk Application
- S3 Bucket Object
- S3 Bucket

Deletion of resources in order is important because they are directly/indirectly dependent of each other for example EC2 instances have EBS Volume attached to it so it we delete the EBS volume first then EC2 instance would get affected from this.

Fig 6 explains the flow of lambda deleting and notifying deleted resources using SES.

- Lambda compares AWS resource with its respective configuration file containing its ID in S3 bucket exclude list
- If AWS Resource is not in exclude list then Lambda deletes the resource and sends notification of deleted resources. Otherwise no action is taken and resources are not deleted.

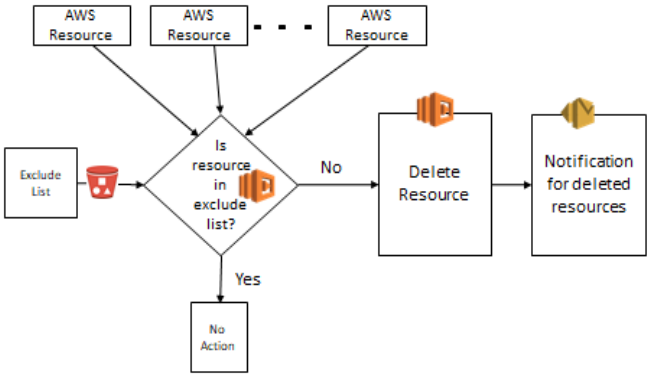

Fig 6: Automated AWS Resource Deletion

Fig 7 shows the internal design and workflow of EC2 and S3 automated notification and automated deletion where, EC2 and S3 are being monitored for their invalid/improper use and notifications are sent for such EC2 Instances and S3 Objects to authenticated person and verified by them according to which automated deletion is done for improper resources.

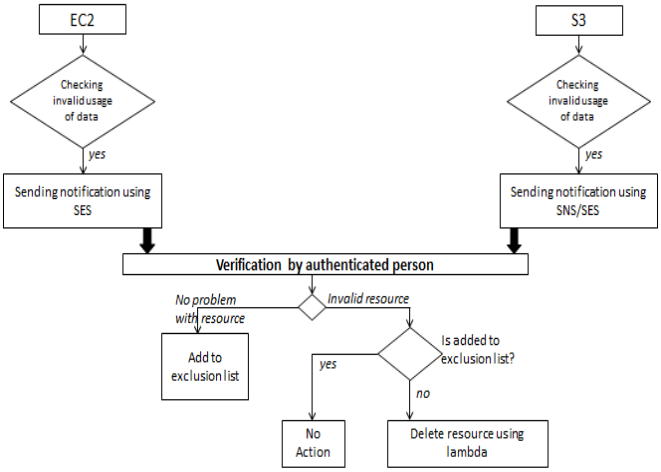

Fig 7: EC2 and S3 Automated Notification and Automated Deletion

DRAWBACKS OF EXISTING METHODS

In a traditional or existing use of AWS within an organization there are some drawbacks and the limitation when there is no monitoring, no automated notification and no management of AWS resources includes:

- *Security:* Security problem because if resource are not monitored for example someone creates an EC2 instance and set its security group as open to world in that case anyone can misuse it, same problem can occur if S3 bucket and object is made public, there can be any confidential information that can leak, security also become weak if anyone left EBS (Elastic Block Storage) or S3 unencrypted.

- *Increase Cost:* Improper maintenance can cause high cost as there can be users who left EC2 in running state, or using improper instance type, or using backups (snapshots) etc which need to monitor and if required delete or change the configuration in right time.
- *Configuration Overhead:* There is also an installation and configuration related overhead, if any developer wants to create application over EC2 then he/she needs to do so many configuration which includes installation of different packages and software which increases the overhead
- *Increase Time:* When we have to deal with configuration overhead, maintenance overhead then time to complete a project also increases.
- *Synchronization Issue:* There can be synchronization problem if developer is creating any project on EC2 and tester on different environment needs to test it.
- *Maintenance Overhead:* With no automation there is maintenance overhead too because otherwise everything needs to be check manually and also there are high chances that resources left unchecked because there are many resources/service that provided by AWS and there are many regions as well
- *Delay in Product Delivery:* If there are so many overheads then it leads to delay in product/application delivery

ADVANTAGES OF A PROPOSED METHOD

- *Security:* AWS environment becomes more secure because security groups, public accessibility and encryption are monitored which helps to ensure the security.
- *Reduce Cost:* Maintenance includes automated deletion of unwanted resources and unnecessarily running resources helps in cost optimization.
- *No Configuration Overhead:* Because of user data and custom AMIs, configuration and installation overhead reduced and it also help in synchronization between different teams.
- *No Maintenance Overhead:* With automated maintenance there is no or less manual maintenance overhead because the only manual step is to take action for excluded list of resources present in S3 bucket i.e. to add/remove the list element according to requirement.
- *Decrease Time:* When we have not to deal with configuration overhead, maintenance overhead then time to complete a project will also decrease.
- Faster delivery of products
- Continuous integration by Jenkins

IV. RESULTS AND DISCUSSION

Custom AMI used for building AWS (Amazon Web Service) instances and verified AWS resources like EC2 and S3 where EC2 instance AMI will be compared with custom AMI and valid AMI used for instance and rest will be deleted, other EC2 comparison includes region, security group, subnet, EBS. Fig 8 shows the result of automated notification for invalid usage of EC2 instance send by lambda using SES where lambda has invoked and scheduled for specific interval by Jenkins.

AWS INVALID INSTANCE REPORT

Instance Id	Image Id	Region	Owner	Instance Type	Subnet ID	Security Group Name	Security Group ID
i-030fedcc63d17512d	ami-5e8bb23b	us-east-2	324568	t2.large	subnet-153f1358	launch-wizard-3	sg-0b5627a9a491ecd49
i-0b6de5e160b8c1bc0	ami-0ff65671a86f061fcd	us-east-2	349587	t2.micro	subnet-153f1358	launch-wizard-11	sg-0632516cf9a34e75f

Fig 8: Automated Notification for invalid data usage in AWS EC2 Instance

S3 objects are verified with respect to accessibility, encryption and type of file and sending notification using SES, SNS for invalid data in AWS Resources and automated notification will be sent for S3 using SNS. Similarly, a SES notification will be sent to admin for unencrypted and public objects of S3.

Fig 9 shows the result of SES notification using lambda for after automated deletion of AWS resources. Similarly, a notification is send before this automated AWS Resource deletion which notify about the resources that will be deleted and the resources that will not be deleted from automated deletion.

Fig 9: SES Notification after Automated Deletion of AWS Resources

V. CONCLUSION AND FUTURE WORK

With enabling automated notification for invalid/improper usage of data in AWS Resources, the infrastructure; application/ product will be more secure, reliable and help in faster delivery of the product. This paper focused on making a standard/pattern/template for EC2 and S3 which helps to provide security, reduced cost and reduced time. This paper also proposed a method to delete unwanted resources and send automated notification about deleted, deleting, excluded resources and for every invalid/improper action. The proposed method also saves from application failure because of custom AMI. With automated notification admin is aware of all the inappropriate actions in the AWS environment and solve them in advance i.e. before the actions took a hazardous shape. The future work is to make this management more secure and automated, to achieve this it is important to identify the improper resources and directly delete them without any interaction with admin. Also to make it secure, maintain the integrity of security groups by a background process that will stop it from being changed to some specific ports and restrict AWS users to create security group with those specific port.

REFERENCES

1. Pedro Álvarez, Sergio Hernandez, Javier Fabra and Joaquin Ezpeleta, *"Cost-driven provisioning and execution of a computing-intensive service on the Amazon EC2, computer science theory, methods and tools"*, the computer journal, Vol.61, Issue.9, pp.1407-1421, 2018
2. Huankai Chen, Frank Z. Wang and Na Helian, "*Entropy4Cloud: Using Entropy-Based Complexity to Optimize Cloud Service Resource Management*", IEEE transaction on emerging topics in computational intelligence, Vol.2, Issue.1, pp.13-24, 2018
3. Xiaolong Liu, Shyan-Ming Yuan, Guo-Heng Luo, Hao-Yu Huang and Paolo Bellavista, *"Cloud Resource Management with Turnaround Time Driven Auto-Scaling"*, IEEE Access, Vol.5, pp.1-10, 2017
4. Daniel Sun, Min Fu, Liming Zhu, Guoqiang Li and Qinghua Lu Member, "*Non-intrusive Anomaly Detection with Streaming Performance Metrics and Logs for DevOps in Public Clouds: A Case Study in AWS"*, IEEE Transactions on Emerging Topics in Computing, Vol.4, Issue.2, pp.278-289, 2016
5. Jan Stanek, and Lukas Kencl, "*Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage*", IEEE Transactions on Dependable and Secure Computing, Vol.15, Issue.4, pp.694-707, 2017
6. Pieter-Jan Maenhaut, Hendrik Moens, Bruno Volckaert, Veerle Ongenae and Filip De Turck, "*Resource Allocation in the Cloud: From Simulation to Experimental Validation*", IEEE 10th International Conference on Cloud Computing, pp.701-704, 2017
7. Maciej Malawski, Adam Gajek, Adam Zima, Bartosz Balis, and Kamil Figiela, "*Serverless execution of scientific workflows: Experiments with HyperFlow, AWS Lambda and Google Cloud Functions*", Future Generation Computer Systems, pp.1-13, 2017
8. Suyel Namasudra, Pinki Roy, "*Time saving protocol for data accessing in cloud computing"*, IET Communications, Vol.11, Issue.10, pp.1558-1565, 2017
9. Nguyen Cong Luong, Ping Wang, Dusit Niyato, Wen Yonggang and Zhu Han, "*Resource Management in Cloud Networking Using Economic Analysis and Pricing Models: A Survey*", IEEE Communications Surveys & Tutorials Vol.19, Issue.2, pp.954-1001, 2017
10. Nicolae Paladi, Christian Gehrmann, and Antonis Michalas, "*Providing User Security Guarantees in Public Infrastructure Clouds*", IEEE Transactions on Cloud Computing, Vol.5, Issue.3, pp.405 – 419, 2016

AUTHORS PROFILE

Pragya Nayak is from Chhatarpur, Madhya Pradesh. She is working as a Cloud Engineer in Broadridge Financial Solutions (India) Private Limited, Bangalore, Karnataka and she works on cloud related tools like Chef, Jenkins, AWS tools, Terraform etc. She has completed her M.Tech in Computer Science and Engineering with Specialization in Cloud Computing from Vellore Institute of Technology, Chennai Campus, Chennai, Tamil Nadu. She has completed her B.Tech in Computer Science and Engineering from Jayoti Vidyapeeth Women University, Jaipur, Rajasthan. Her area of interest for research and publication are in cloud computing related topics including cloud security, cloud storage, load balancing etc.

Dr. Jenila Livingston L. M., Associate Professor, is with School of Computing Science and Engineering, VIT University, Chennai, TN 600127 INDIA. (e-mail: jenila.lm@vit.ac.in). She has completed her PhD in Faculty of Engineering from National Institute of Technical Teachers' Training and Research (NITTTR), Government of India, Chennai and Masters Degree in Computer Science and Engineering from Anna University, India. She has nearly 15 years of experience in Teaching and Research and keenly interested in the areas of eLearning, Engineering Education, Artificial Intelligence, Soft Computing, Data Analytics, Internet & Web Programming, Data Base Systems and Data Structures & Algorithms.

.