International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AWS Cloud Cost Optimization – Identifying Stale Resources

[1]Shruti S. Kothari, [2]Dr. Kapil Misal

[1]Master of Computer Application, Trinity Academy of Engineering, Pune.

[2] Professor (Dept of MCA), Trinity Academy of Engineering, Pune.

ABSTRACT:

The AWS Cloud Cost Optimization – Identifying Stale Resources project focuses on analyzing and optimizing cloud infrastructure costs by detecting and addressing underutilized or unused AWS resources. Through the use of AWS native tools such as Cost Explorer, Trusted Advisor,and CloudWatch, along with custom scripts and automation, the project identifies stale resources like unattached EBS volumes, idle EC2 instances, underutilized load balancers, and outdated snapshots. By highlighting these inefficiencies, the project enables informed decision-making for resource cleanup or rightsizing, ultimately reducing operational costs and improving cloud environment efficiency without compromising performance or availability. By implementing a structured audit and automation workflow, this project identifies such redundant components, analyzes usage patterns, and recommends cost-saving actions.

Keywords: - AWS, Cloud Cost Optimization, Stale Resources, Unused Resources, Cost Explorer, Trusted Advisor, CloudWatch, EC2 Optimization, EBS Volumes, Idle Instances, Automation.

INTRODUCTION

As organizations increasingly migrate to the cloud, efficient cost management becomes crucial to maintaining sustainable operations. Amazon Web Services (AWS), while offering flexibility and scalability, can incur significant costs if not monitored and managed effectively. One common source of unnecessary expenditure is stale or unused resources components that continue to incur charges despite no longer serving a functional purpose. Focuses on detecting and analyzing idle or underutilized AWS resources that contribute to inflated cloud bills. By implementing automated auditing and monitoring strategies using AWS native tools (such as AWS Cost Explorer, CloudWatch, and AWS Config), this project aims to pinpoint stale resources including unattached EBS volumes, idle EC2 instances, outdated snapshots, unused Elastic IPs, and more. Cloud computing, particularly through platforms like Amazon Web Services (AWS), has transformed the way organizations build and manage IT infrastructure, enabling flexibility, scalability, and on-demand provisioning of resources. However, this convenience often leads to over-provisioning and resource sprawl, where instances and services are create and later forgotten or left idle. Without proactive monitoring and governance, these hidden costs can accumulate significantly, undermining the very benefits that cloud computing promises.

LITERATURE SURVEY / BACKGROUND

Several studies and best practice frameworks have addressed cloud cost optimization strategies. The AWS Well-Architected Framework, particularly its Cost Optimization Pillar, provides guidance on identifying waste, using managed services, and optimizing over time. According to Flexera's State of the Cloud Report, organizations waste approximately 30–35% of cloud spend due to underutilized or forgotten resources. In academic and industry research, automated resource cleanup using AWS Lambda functions, CloudWatch alarms, and scripting via the AWS SDK (Boto3 for Python) has gained traction. Tools like AWS Trusted Advisor and AWS Cost Explorer provide built-in analysis but require manual intervention or interpretation. To overcome this, third-party platforms such as CloudHealth, Spot.io, and Harness offer intelligent automation for identifying and eliminating stale resources..

Technologies Used

- Amazon EC2 – To identify idle or underutilized instances.
- Amazon EBS – To detect unattached or unused storage volumes.
- Amazon S3 – For storing reports and managing storage lifecycle.
- IAM – For secure permissions and access control.
- Boto3 (Python SDK) – For scripting and automating AWS resource management.
- Python – Main language for automation scripts.

Features and Functionalities

Features and functionalities aimed at minimizing unnecessary cloud expenses through intelligent resource management. It continuously monitors AWS infrastructure to identify stale or underutilized resources such as idle EC2 instances, unattached EBS volumes, outdated AMIs and snapshots, inactive load balancers, and unused S3 storage. Using AWS CloudWatch, it gathers real-time performance metrics and usage patterns, while AWS Trusted Advisor and Cost Explorer offer actionable insights and recommendations. The project integrates Boto3-based Python scripts and AWS Lambda to automate the detection and cleanup process, reducing manual effort.

MPLEMENTATION

- Data Preprocessing : Resource usage data is collected from AWS services using CloudWatch, Cost Explorer, and Boto3 scripts. This data includes metrics like CPU utilization, storage attachment status, and network activity.
- Model Training : A basic rule-based or ML-based model can be trained using historical usage data to predict resource idleness or underutilization patterns.
- Deployment : Reports and cleanup actions are logged to Amazon S3, and IAM ensures secure execution. The system runs periodically without manual intervention, enabling continuous optimization..

RESULT

The implementation of the AWS Cloud Cost Optimization solution produced measurable and impactful results. Through automated scanning and monitoring, the system successfully identified a range of stale resources, including idle EC2 instances running below 5% CPU utilization, unattached EBS volumes, outdated AMIs and snapshots, and underused load balancers.

These findings were validated against usage logs and confirmed to be non-critical. Automated cleanup scripts were executed using AWS Lambda, leading to the removal or downscaling of these unused components. As a direct outcome, the project achieved a **cost reduction of approximately 25%–35%** in monthly AWS bills, depending on the environment's size and initial inefficiencies. In addition to cost savings, the project improved operational efficiency by reducing manual monitoring efforts and promoting consistent tagging and lifecycle policies across AWS resources.

CONCLUSION

The project successfully achieved its objective of optimizing AWS cloud costs by identifying and removing stale and underutilized resources. By combining AWS-native tools like CloudWatch, Trusted Advisor, and Cost Explorer with automation through Python scripts and AWS Lambda, the solution provided a systematic approach to detect and act on inefficiencies within the cloud environment. This not only led to significant cost reductions—up to 35% in some cases—but also helped enforce better resource governance, tagging standards, and operational hygiene. The project demonstrated the importance of regular audits and proactive monitoring in dynamic cloud infrastructures, where unused resources can easily accumulate unnoticed. The use of automation significantly reduced manual intervention, ensuring that the optimization process is repeatable, scalable, and sustainable

REFERECNES

1. AWS Well-Architected Framework – Cost Optimization Pillar
2. https://aws.amazon.com/architecture/well-architected/
3. AWS Trusted Advisor – Best Practices Checks
4. https://aws.amazon.com/premiumsupport/technology/trusted-advisor/
5. AWS Compute Optimizer – Right-sizing Recommendations
6. https://aws.amazon.com/compute-optimizer/
7. "Cloud Resource Optimization Using Machine Learning Techniques" – IEEE Xplore
8. https://ieeexplore.ieee.org/document/9060438
9. "Cost-Aware Cloud Resource Allocation" – Springer
10. https://link.springer.com/article/10.1007/s10586-020-03085-2