A Cost-Effective And Practical Solution For AWS Resources Management With Usage Visualization

MSc Research Project

Research in Computing CA2

Bhushan Chaudhari

Student ID: x21165840

School of Computing

National College of Ireland

Supervisor: Dr Puneet Gupta

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Bhushan Chaudhari		
Student ID:	x21165840		
Program:	Msc in Cloud Computing	**Year:**	2023
Module:	Research in Computing CA2		
Supervisor:	Dr Puneet Gupta		
Submission Due Date:	14 August 2023		
Project Title:	A Cost-Effective And Practical Solution For AWS Resources Management With Usage Visualization		

Word Count: ………………………………………. **Page Count** 27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Bhushan Chaudhari

Date: 13 August 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only

Signature:	
Date:	
Penalty Applied (if applicable):	

Cost-Effective And Practical Solution For AWS Resources Management With Usage Visualization

Bhushan Chaudhari

x21165840

1. Abstract

Cloud computing is a revolutionary paradigm that enables users to access and utilize a vast array of computing resources and services over the internet. Instead of relying on locally installed hardware and software, cloud computing allows individuals and organizations to access data storage, applications, and processing power through remote servers hosted by third-party providers. Users don't have to own and manage physical hardware and software, cloud computing offers a pay-as-you-go model, (Ibrahimi, 2017) allowing organizations to scale resources dynamically based on demand. Amazon Web Services (AWS) is one of the leading cloud computing service providers, offering a vast array of cloud-based solutions and services .With AWS's pay-as-you-go pricing model, users can avoid upfront infrastructure costs and only pay for the resources they consume. While cloud computing eliminates upfront hardware expenses, the recurring subscription costs for cloud services can add up over time. Organizations may unknowingly accrue additional costs due to idle or underutilized resources in the cloud. The pay-as-you-go model means that resources left running without being actively used can lead to wasted expenditure.

This abstract presents a comprehensive analysis of the factors contributing to cost effectiveness in cloud computing, highlighting the key strategies and best practices that can be

implemented to optimize resource allocation and expenditure**.** In this study, we'll review and put into practice a system that suggests allocating and de-allocating resources in accordance with usage. The system is capable of offering a single dashboard for controlling all of the accounts in an AWS account as well as all of the resources allotted inside those accounts.
Keywords: Amazon Web services (AWS), Cloud Watch, Cost Optimization, Resource Allocation, Server, Central dashboard, Time Zone.

2. Introduction

Cloud computing is an advanced technology that enables users to gain access to and employ computing resources, such as servers, storage, databases, networking, software, and analytics, via the internet. By means of renting or leasing these resources from a cloud service provider, businesses and individuals can attain a cost-effective and adaptable solution, without the need to own or maintain physical hardware. In the realm of cloud computing, users can provision and manage computing resources, such as virtual machines and storage, without any human intervention from the cloud service provider, thereby empowering users to access and configure resources as required, and thereby reducing the need for direct interaction with support staff. Cloud providers apply a multi-tenant model, wherein computing resources are shared among several users. Nonetheless, the data and applications of each user are kept isolated from others, thus ensuring utmost security and privacy. This pooling of resources allows for optimal utilization and scalability, thereby minimizing waste and enhancing performance. It furnishes the ability to respond to shifting demands, while lowering operational complexity and costs.

Amazon Web Services (AWS) is a comprehensive and widely used cloud computing platform offered by Amazon. It was one of the first major cloud providers and remains a leader in the industry. AWS offers an extensive range of services, catering to various business needs. In cases where the organization's resource usage is consistently high, the accumulated subscription fees might surpass the cost of owning and operating physical infrastructure in the long term

In order to carry out the study, we have settled on the following research questions:

1. How can one effectively monitor the utilization of Amazon Web Services (AWS) assets spanning multiple accounts and recommend the retirement of redundant assets?
2. How can one more effectively handle multiple subsidiary accounts within an Amazon Web Services (AWS) organizational account?
3. How might we amalgamate all of our cloud-based resources into a solitary dashboard whereby we can oversee their performance and allocate authorizations?
4. How can one initiate and terminate cloud-based assets on a predetermined schedule, while also considering variances in time zones?

3. Related Work

There are very few works that have been done in the research direction that we embark on which shows how traditional cloud computing can be enhanced to give performance and results but somehow lack in one or the other criteria.

Some research papers which show similar development are:

1. A "No Data Center" Solution to Cloud Computing

The Credit Union Cloud approach (CUCM) is one such cloud provisioning approach that tries to tap into the overabundance of idle/underutilized computers for cloud service provisioning, whereas traditional practice relies on specialized data centers. CU clouds operate on existing infrastructures with extra capacity that are not specifically designed to enable Cloud Computing. These PCs are not specialized resources for the cloud infrastructure; instead, they are utilized as usual by their intended users, such as running a word processor or browsing the web (referred to as local/native applications). (Mengistu, 2017)CU cloud enables an organization's PCs to join the "cloud credit union" and contribute underused resources (such as CPU cycles or disk space) to the union's cloud resource pool.

Disadvantage:

The requirement of CUCM is to have a robust, dynamic and efficient resource management and provisioning mechanism.

2. **C-Cloud: A Cost-Efficient Reliable Cloud of Surplus Computing Resources**

The article introduces C-CLOUD, an open-source cloud platform for leasing computing resources, including non-cloud assets (computers, laptops, enterprise servers, and clusters that are not part of any cloud infrastructure). (Dutta, 2014)C-CLOUD allows for the rental of massive amounts of spare computer resources to cloud users. Such resource sharing allows resource owners to profit from idle resources while providing cloud customers with a cost-effective alternative to huge cloud providers.

Disadvantage:

C-CLOUD faces two major challenges: assuring SLA and dependability of reservations made over diverse resources, and providing an acceptable mechanism for motivating resource sharing.

3. **Cloud Bursting Scheduler for Cost Efficiency:**

The rising use of cloud computing is their flexibility and pay-as-you-go (PAYG) pricing. While many businesses use public clouds like Amazon Web Services' Elastic Compute Cloud (EC2) for their full ICT solution. (Lian, 2017)In this study, we introduce the Cloud Bursting Scheduler (CBS), a new cloud bursting algorithm that dynamically distributes workloads and jobs between private systems and public clouds with the goal of reducing the cost of carrying out those jobs. CBS specifically considers the time-varying electricity charges with private clouds and the time-invariant rental fee of various public clouds when offloading jobs to public clouds in an effort to maximize the cost-to-performance ratio.

Disadvantage:

In this work, the attempt to share a resource by offloading jobs to a public cloud by a cloud bursting scheduler will not only compromise the security of the shared resource but also compromise data privacy.

4. **One Quantifiable Security Evaluation Model For Cloud Computing Platform**

As the technology for cloud computing develops, it has become a usual method for many people to create their own information infrastructure. Hyperscale datacenters are used by rapidly expanding businesses like AMAZON AWS, Microsoft AZURE, Ali Cloud, and others to provide cloud services. (A. Sun, 2018) The cloud technology has given us access to relatively affordable services and operational conveniences, but it has also led to a fragmented user information infrastructure. Without knowing whether the cloud services they use are secure and whether their data can be securely kept in different clouds, cloud clients cannot understand the complete security picture and effectively handle security issues. For the cloud computing infrastructure, the study creates a security monitoring engine, a security recovery engine, and a security quantified evaluation model.

Disadvantage:

Currently, the study uses a single cloud environment with a limited set of APIs to handle security, but it fails to use multiple clouds with all of the APIs and resource management.

5. **Cloud Computing: A Risk Assessment Model**

Cloud computing has numerous qualities, including on-demand self-service, widespread network access, resource pooling, quick adaptability. Despite the significant economic advantages that cloud computing provides for businesses, risk assessment is difficult due to the specific security concerns that have been developed in a cloud environment. (Cheriet, 2014)Customers of cloud services require security for their cloud applications against online threats. In this paper, a framework for evaluating the security risks connected to cloud computing platforms is proposed.

Disadvantage:

The limitation of the model is that it is not fully automated risk assessment models in cloud and also cost will eventually be increased in implementing this manual process.

6. **A New Energy Efficient VM Scheduling Algorithm for Cloud Computing based on Dynamic Programming**

Cloud computing is a modern computing infrastructure that allows users to access a wide range of computer resources on demand. It is based on a distributed system that uses virtualization and network technology to deliver resources. (K. Zhang, 2017)With cloud computing, users can easily request and release computing, storage, network, and other resources, just like they would use water and energy. However, as the size of cloud computing data centers has grown, the energy usage of servers and other infrastructure has also increased, leading to higher operating expenses and significant environmental issues. The author focuses on choosing virtual machines (VMs) for a host that is overloaded in a cloud computing environment in order to avoid having to pay for several virtual machines.

Disadvantage:

The paper lacks in scheduling task and allocating the resources as per demand or usage but has only focused on virtual machine.

7. **Cloud Computing: Comparison and Analysis of Cloud Service Providers—AWS, Microsoft and Google**

When utilizing a cloud computing platform, IT resources can be accessed and released on an as-needed basis. This service is available to both businesses and individuals who can benefit from utilizing services such as IaaS, PaaS, SaaS, DaaS, and more, provided by cloud service providers who offer pay-as-you-go options. The three leading vendors based on market share are Amazon, Microsoft, and Google. (Tripathi, 2020) In this paper, we'll examine a few of the tools that AWS, Azure, and GCP, the top three providers of cloud computing services, have to offer. These tools include compute, storage space management, and performance. There is no uniform standard, and the growth of these providers is happening in parallel directions with many of them concentrating on processing power and offering end users services for CPU, storage, databases, and networking. Some service providers prioritize lowering the cost of the service, while others prioritize uninterrupted service delivery and service scalability.

Disadvantage:

After comparing the all the providers, it seems that the resource managing tool is inefficient of managing the unused resources. Their main focus is on providing the Open source platform and public cloud but somehow lacks in cost management which will be incurred by the customers.

8. Cost-Benefit Analysis of Cloud Computing in Education Using the Base Cost Estimation Model

Schools and institutions are under more pressure to produce more with fewer resources as a result of COVID-19 and the desire for higher quality education. To help with this problem, educational institutions can use cloud computing to boost capacity while reducing Total Cost of Ownership (TCO) and avoiding costly infrastructure investments. (J. F. Salindeho, 2021)In order to give the institution a better understanding of how much the data center actually costs and whether moving to the cloud will reduce those expenses, the researcher plans to examine the costs associated with data centers at universities and compare those costs to pricing estimates generated for the cloud. When comparing the costs of two cloud providers, Amazon Web Services and Microsoft Azure, the cloud data center costs will be estimated based on the data analysis of the servers' specs. The figures for the costs of the physical data center, the AWS cloud, and the Azure cloud are then compared to see which option has the lowest costs.

Disadvantage:

Interestingly, the highest cost among the all alternatives is the AWS cloud. Even universities fail to calculate where to actual use the cloud service and where to stick to physical machine so that the cost is lesser.

9. Optimization of Resource Provisioning Cost in Cloud Computing

Two computing resource provisioning options, referred to as reservation and on-demand plans, are available to cloud computing users from cloud providers. Since cloud consumers must pay the provider in advance, the cost of using computing resources provided by reservation plans is typically lower than that provided by on-demand plans. The consumer can lower the overall resource provisioning cost using the reservation plan. (S. Chaisiri, 2012)Due to the unpredictability surrounding consumer demand and resource provider costs, it is challenging to make the best advance reservations for resources. By developing a stochastic programming model, an optimal cloud resource provisioning (OCRP) method is put forth as a solution to this issue. The OCRP algorithm can supply computing resources for usage in a long-term plan as well as various provisioning stages.

Disadvantage:

It can surely solves the problem of the long term users and short term user with different available plan, which in turn can cut the cost of resources used but still the cost incurred by the resources which are purchased and not been used will still persist.

4. Research Methodology

4.1 Proposed system:

The primary objective of this research is to investigate and evaluate the effectiveness of using AWS Instance Scheduler and RDS Scheduler for cost optimization in cloud environments. The study aims to identify and decommission least used machines and functions to reduce cloud infrastructure costs while maintaining reliability and performance.

Specifically, the research aims to:

1. **Analyze AWS Instance Scheduler:** Investigate the capabilities of AWS Instance Scheduler in automating the start and stop times of instances based on workload patterns

and demand. Evaluate its effectiveness in reducing costs by preventing the use of unnecessary resources during idle periods.

2. **Identify Least Used Machines and Functions:** Develop methodologies to identify instances, virtual machines, and functions that experience consistently low utilization. This involves analyzing historical usage data, identifying usage patterns, and setting appropriate utilization thresholds.

3. **Cost Optimization Algorithm:** Propose cost optimization strategies for decommissioning identified least used machines and functions. Evaluate the impact of these strategies on cost reduction while considering their effects on reliability and performance.

4. **AWS Metric for Monitoring Resources:** AWS Metrics are quantitative data collected and generated by various AWS services that provide valuable insights into the health, performance, and behavior of resources and applications running within the AWS cloud environment. With the help of AWS metrics, we can measure CPU utilization, network traffic, and request rates. Additionally, we will define custom metrics to monitor specific application-specific parameters.

5. **RDS Scheduler For Decommissioning Of Resources :** RDS Scheduler will be used to define specific start and stop times for RDS instances, helping them optimize costs and resources for non-production and development environments that do not require 24/7 availability. We will set the instances to run only during business hours or specific time frames, reducing costs incurred during idle periods.

6. **Resource Allocation Based On Different Time Zone:** The methodology supports scheduling instances in multiple time zones, catering to global teams and diverse business requirements. We will create recurring schedules, allowing them to establish consistent on/off patterns for different days of the week or specific intervals.

4.2 Data Collection:

The research will employ the textual data patterns to identify what went wrong which lead to the higher cost of using the AWS cloud by the users. The research will collect relevant datasets that provide insights into the cost before and after implementing the cost optimization strategies.

Here are some parameters which will be checked before and after implementing the new strategies:

a. Instance Scheduling Data:

Data is captured related to instance scheduling activities, including start and stop times of instances. Record instances' uptime and downtime to calculate cost savings achieved through scheduling.

b. Data on Least Used Instances and Functions:

To demonstrate decommissioning least used resources, we have collect data on instances or functions with consistently low utilization over time.

c. Performance and Reliability Metrics:

We have gathered data on application performance and reliability metrics (e.g., response time, uptime, error rates) before and after optimization to ensure that performance is not adversely affected.

d. Auto scaling Events Data:

As we are using AWS metrics, we have collected the data on scaling events, such as when instances are launched or terminated based on workload demand. We have collected workload data to understand the demand patterns of your application in different time zone. This can include information about incoming requests, data processing, and any time-sensitive operations.

5. Design Specification

5.1 Functional Requirements

1. **User registration**

   Cloud cost optimization project will offer Software as a service to optimize cloud cost for various AWS account holders. Account holder need to register using access key and secret for their account. The AWS credentials (access key and secret) will be used to connect to AWS API and fetch all details on various AWS components.

2. **Fetch AWS resources**

   Cloud cost optimization project connects to all user's AWS accounts and fetches all instances, volumes and lambda functions. All details are saved in database for showing on the UI screens.

3. **Fetch AWS Cloudwatch Metrics**

   Cloud cost optimization project fetches Metrix for all resources and loads them to database.

Sr.No	Resource	Metrix
1	EC2	CPUUtilization, NetworkIn,NetworkOut, DiskReadBytes, DiskWriteBytes, DiskReadOps,DiskWriteOps
2	EBS	VolumeReadBytes,VolumeIdleTime, VolumeWriteOps, VolumeQueueLength,VolumeWriteBytes,VolumeReadOps, BurstBalance,VolumeTotalReadTime,VolumeTotalWriteTime

4. **Resource Optimization Decision using Support Vector Machine**

   For any AWS account, the base computing engine used is EC2. If we want to find out resource utilization and effectiveness for EC2, then its day wise usage needs to analyzed for last 30 days.

   Metrix include CPU(95 percentile), RAM Used(95 percentile), NetworkOut(p95), VolumeWrite(p95) InstanceType

   Instance types are really important in taking decision to allocate or deallocate a resource. Bigger instances might have higher threshold for CPU usage due to increased cost of instance, whereas smaller instances might have lower threshold values as they do not result in much cost for the account.

Sr.No	Instance	vCPU*	CPU Credits / hour	Mem (GiB)	Storage	Network Performance
1	t2.nano	1	3	0.5	EBS-Only	Low
2	t2.micro	1	6	1	EBS-Only	Low to Moderate
3	t2.small	1	12	2	EBS-Only	Low to Moderate
4	t2.medium	2	24	4	EBS-Only	Low to Moderate
5	t2.large	2	36	8	EBS-Only	Low to Moderate
6	t2.xlarge	4	54	16	EBS-Only	Moderate
7	t2.2xlarge	8	81	32	EBS-Only	Moderate

These metrics are then supplied to Support vector machine to find out whether to allocate or de-allocate a specific machine.

Dataset for SVM is updated based on user's personalized preferences as per their organizations.

5. **Instance Scheduler for EC2**

Instance scheduler allows user to define ON/OFF schedules for their compute resources. Each resource is timed based on the user's time zone, user can define start time, end time, days (weekdays or weekends or custom) to schedule the resource. Scheduler runs every 10 minutes to check the schedule and applies ON/OFF action on ec2 resources.

5.2 Non-Functional requirements

5.2.1 Performance:

- **Responsiveness:** The scheduler should respond promptly to user-initiated actions, such as creating or modifying schedules.
- **Scalability:** The scheduler must handle a growing number of instances and schedules without significant performance degradation.
- **Efficiency:** The resource utilization of the scheduler itself should be optimized to minimize overhead and operational costs.

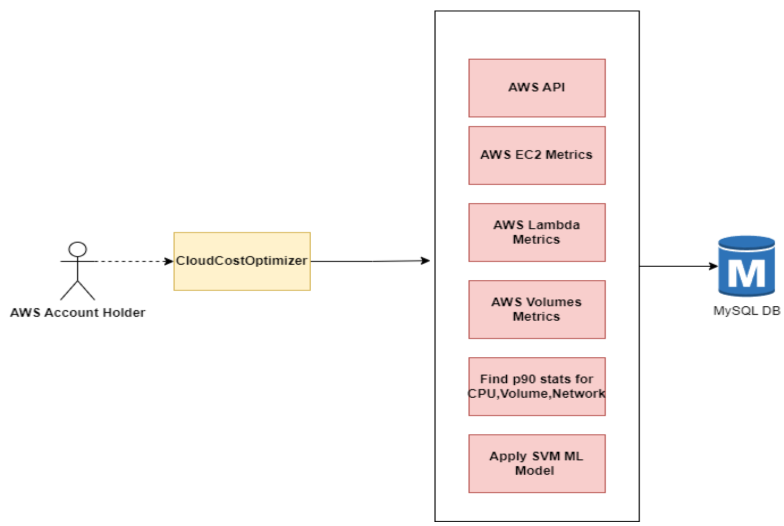
5.2.2 Reliability:

- **Availability:** The cost optimization solution should be highly available and accessible to users when needed.
- **Error Handling:** The solution should gracefully handle errors, exceptions, and failures to prevent data loss and ensure continuous operation. Instance scheduler should ON/OFF timings should occur in a day
- **Data Integrity:** Ensure the accuracy and integrity of cost data throughout the optimization process.

5.2.3 Performance Metrics:

- **Monitoring:** Provide monitoring capabilities to track the cost optimization solution's performance and resource usage.
- **Reporting:** Offer comprehensive cost reports and analysis to help users assess the effectiveness of optimization efforts.

5.3 System Architecture

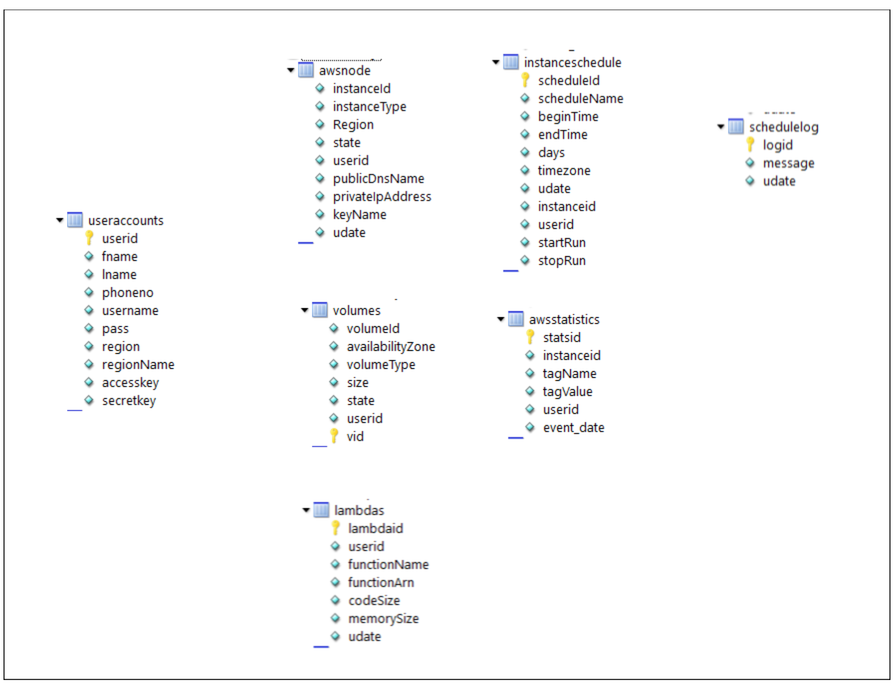


The System architecture consists of

- User registration – User registration consists of user name,password,aws account secret and access key. AWS account holders will use the system to reduce on their resource utilization bills and find out resources that needs attention. Access key and secret are used to connect to aws account and fetch information using AWS SDK.

- AWS Cloud watch metrics – it is used to identify usage of a particular compute resource for last 30 days. We use percentile 90 to find out what was the most used value of the metric. Different metrics are fetched as mentioned in 5.1.3 and stored in mysql database for later processing. For each day, 1 value is fetched for a metric and kept in database.
- Lambda – lambda is widely used compute resource, we fetch all lambdas for user and find out lambdas that are occasionally used.
- EC2 Resources and Volumes – EC2 is the backbone of aws and used as a base compute engine for almost all services. We fetch all ec2 instances and their metrics to find out least used machine.
- Mysql RDS - RDS is used to store all information about users and metrics

5.4 Data Model



Database consists of

a. Useraccounts
b. Awsnode
c. Volumes
d. Lambdas
e. Instanceschedule
f. Awsstatistics
g. Schedulelog

5.5 Algorithms and Methods

Support Vector Machines (SVM) is a machine learning algorithm used for classification of resources for cost

1. Data Collection:

   In order to optimize cloud resources, we have collected historical data of usage and de-allocation strategy for EC2 machines. The metrics used are CPU utilization, memory utilization, network used, instance type (1 t2.nano-7 t2.2xlarge) and decision (1 for de-allocation and 0 for allocation)

```
@relation test
@ATTRIBUTE "% CPU Used for a month" NUMERIC
@ATTRIBUTE "% RAM Used for a month" NUMERIC
@ATTRIBUTE "% Volume Used for a month" NUMERIC
@ATTRIBUTE "Network used for a month" NUMERIC
@ATTRIBUTE "InstanceType" NUMERIC
@ATTRIBUTE "Decision 1-deallocate 0-allocate" {0,1}
@data
10,2,0,10,1,1
30,5,2,10,1,0
40,2,10,5,1,0
5,5,0,5,2,1
10,1,0,10,2,1
30,1,2,10,2,0
```

2. Data Preprocessing

   All metrics are converted to 90 percentile values and then used for decision making. Textual attributes like InstanceType is converted to numeric using instance type table mentioned in 5.1.4

3. Model Training

Above data is trained using SVM model. 80% of data is used for training and 20% of the data is used for testing. As the dataset size is very small and problem is straight forward, we have used SVM model which provides close 95% accuracy.

4. Model Evaluation

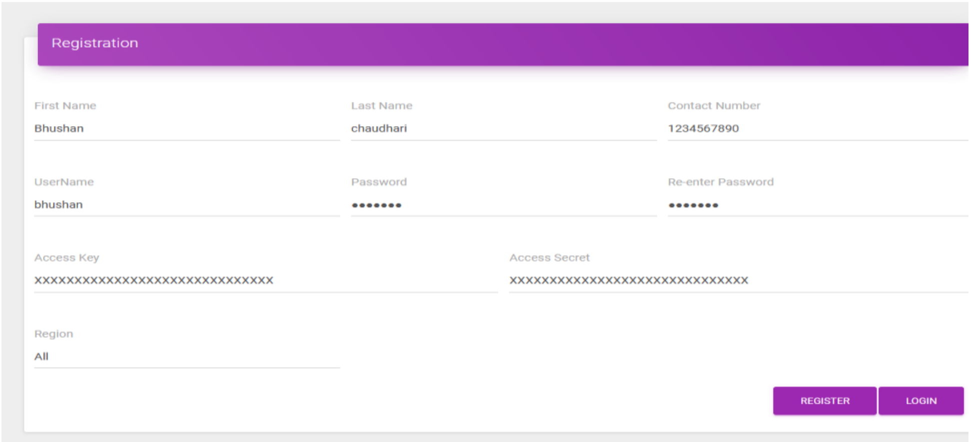
   Evaluate the trained model's performance using appropriate evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE). You want the model to accurately predict costs based on resource metrics.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1	0.125	0.889	1	0.941	0.882	0.938	0.889	0
0.875	0	1	0.875	0.933	0.882	0.938	0.938	1
0.938	0.063	0.944	0.938	0.937	0.882	0.938	0.913	

6. Implementation

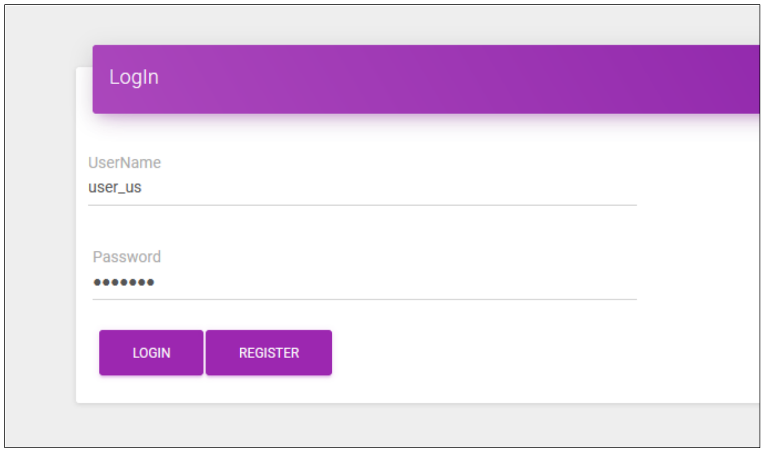
6.1 Signup Page

Signup page allows user to enter their access key and secret to access AWS account resources.



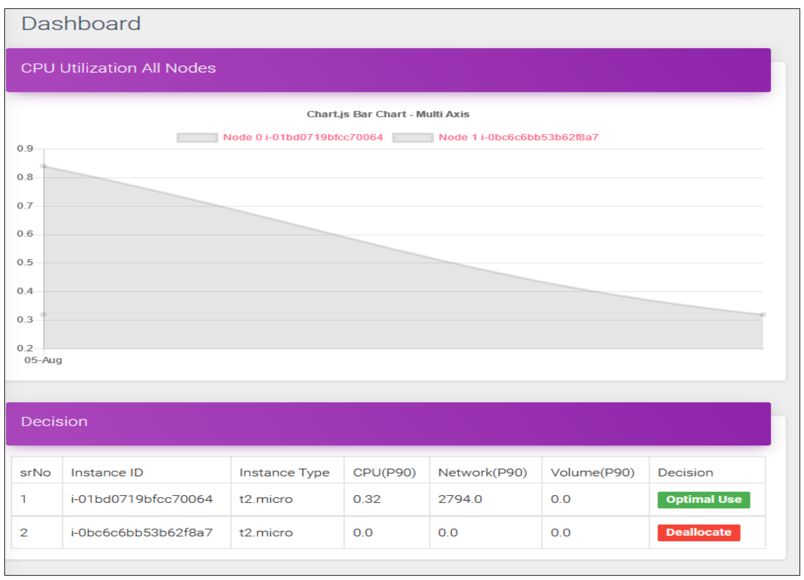
6.2 Login Page

Login page allows user to view their account information.



6.3 Resource Dashboard with Decision

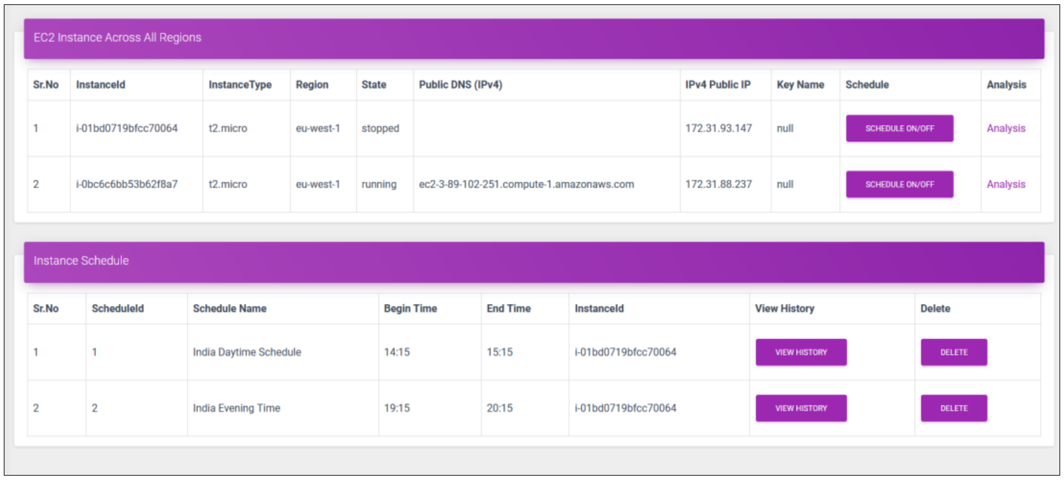
EC2 is the backbone of AWS and used as a base compute engine for almost all services. We fetch all ec2 instances and their metrics to find out least used machine.

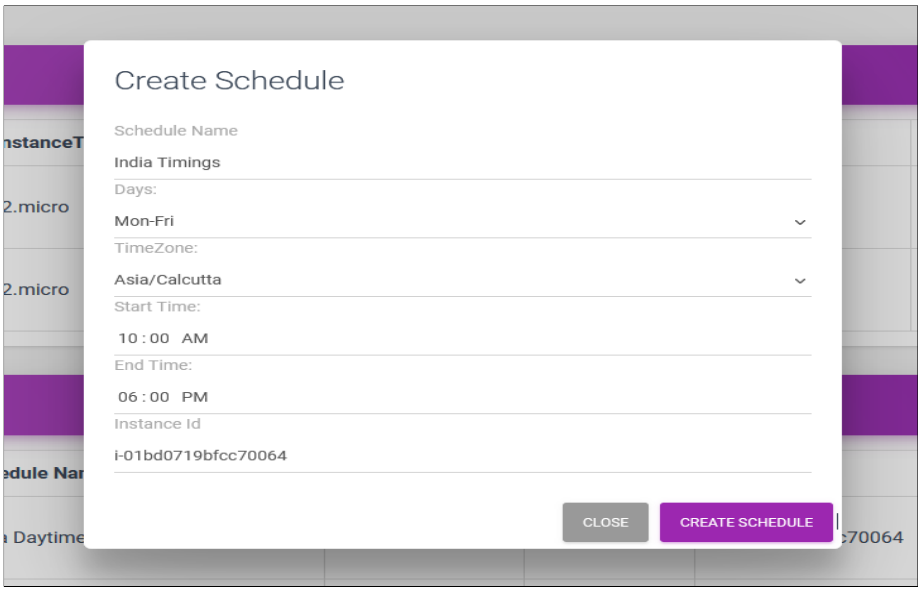


srNo	Instance ID	Instance Type	CPU(P90)	Network(P90)	Volume(P90)	Decision
1	i-01bd0719bfcc70064	t2.micro	0.32	2794.0	0.0	Optimal Use
2	i-0bc6c6bb53b62f8a7	t2.micro	0.0	0.0	0.0	Deallocate

6.4 Instance Scheduler

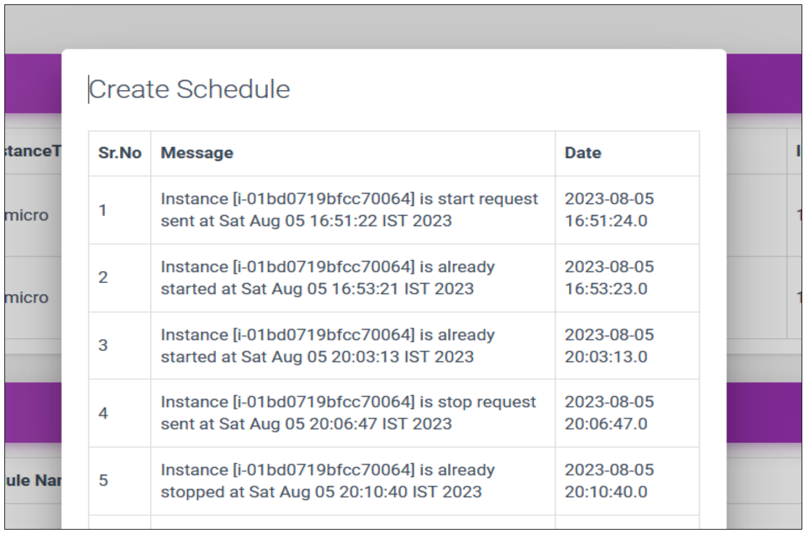
On instance scheduler user can select when to start/stop the instance. It works as per the user's time zone. User can select days and timings as per 24 hour clock.





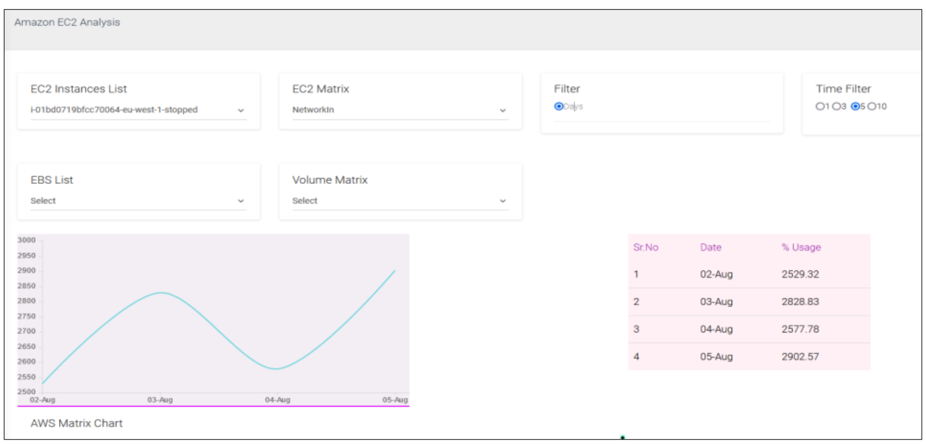
6.5 Schedule History

All instance scheduler history is recorded and stored in scheduledlog table.



6.6 AWS metrics explorer

Resource utilization can be viewed using this screen.

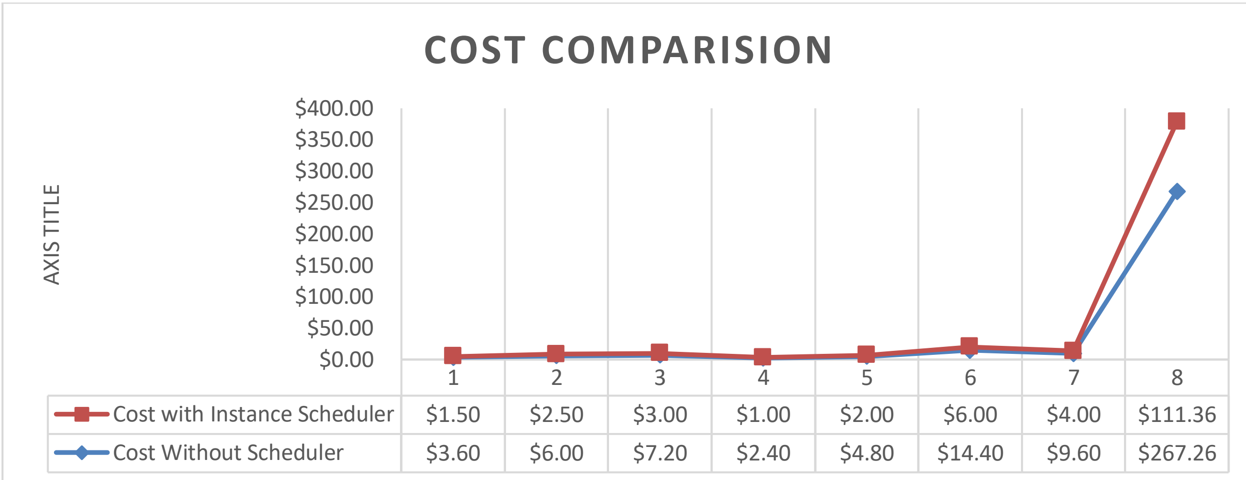


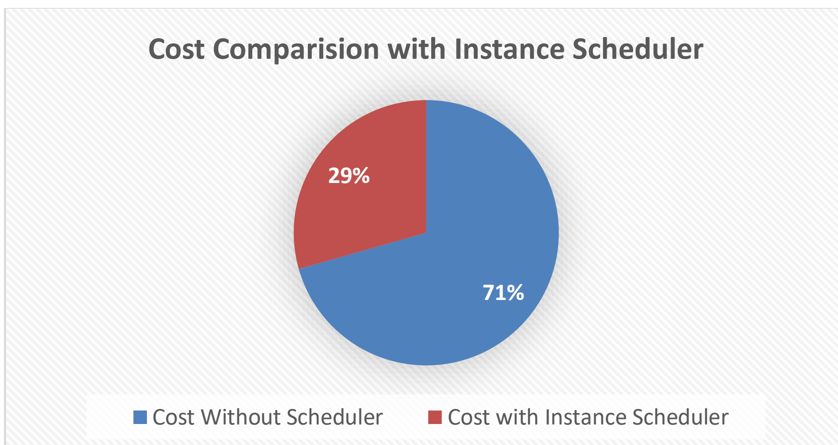
7. Evaluation

Based on the provided data for instances with and without an instance scheduler, here's an analysis of the cost optimization results:

1. **Cost Savings:** Implementing the instance scheduler has resulted in significant cost savings for each instance. The "Cost with Instance Scheduler" is considerably lower than the "Cost Without Scheduler" for each instance.
2. **Variable Costs:** The cost of instances varies based on factors like instance type, age of the instance, and timezone. Instances with longer ages tend to have higher costs.
3. **Total Cost Savings:** The total cost savings across all instances due to using the instance scheduler is $183.90 ($315.26 - $131.36).
4. **Impact on Different Instance Types:** The cost reduction impact is observed across various instance types, from t2.micro to t2.2xlarge.
5. **Significant Savings for Larger Instances:** The cost savings are particularly significant for larger instances like t2.2xlarge, where the "Cost with Instance Scheduler" is substantially lower than the "Cost Without Scheduler."
6. **Importance of Instance Age:** The age of the instance plays a role in cost optimization. Newer instances tend to have lower costs, and the instance scheduler helps in optimizing their usage.
7. **Timezone Impact:** The timezone of the instances also impacts the cost optimization, with instances in different timezones showing varying levels of savings.
8. **Holistic Approach:** The instance scheduler's impact is not only about direct cost savings but also about aligning resource availability with actual workload demands.

Sr.No	Instance Id	Instance type	Timezone	Age of instance	Instance cost per hour	Cost Without Scheduler	Cost with Instance Scheduler
1	i-01bd0719bfcc70064	t2.micro	ap-south-1	15	$0.01	$3.60	$1.50
2	i-01bd0719bfcc70064	t2.micro	ap-south-1	25	$0.01	$6.00	$2.50
3	i-01bd0719bfcc70064	t2.micro	ap-south-1	30	$0.01	$7.20	$3.00
4	i-01bd0719bfcc70064	t2.micro	ap-south-1	10	$0.01	$2.40	$1.00
5	i-01bd0719bfcc70064	t2.micro	ap-south-1	20	$0.01	$4.80	$2.00
6	i-0bc6c6bb53b62f8a7	t2.small	eu-west-1	30	$0.02	$14.40	$6.00
7	i-abc6c6bb53b62f8a7	t2.2xlarge	eu-west-1	20	$0.02	$9.60	$4.00
8	i-abc6c6bb53b62f8a7	t2.2xlarge	eu-west-1	30	$0.37	$267.26	$111.36
					Total	$315.26	$131.36





Following results are obtained for machine learning model used for cost optimization in the project. Its providing close to 94% accuracy for both the classes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1	0.125	0.889	1	0.941	0.882	0.938	0.889	0
0.875	0	1	0.875	0.933	0.882	0.938	0.938	1
0.938	0.063	0.944	0.938	0.937	0.882	0.938	0.913	

Correctly Classified Instances	15	93.75%
Incorrectly Classified Instances	1	6.25%
Mean absolute error	0.0625	
Root mean squared error	0.25	
Relative absolute error	12.3188	
Root relative squared error	49.2598	
Total Number of Instances	16	

8. Conclusion And Future Work

Cost optimization in AWS is a pressing need for organizations as they strive to maximize their cloud investments while maintaining efficiency, performance, and reliability. As the adoption of cloud services continues to grow, so does the complexity of managing cloud resources and associated costs.In today's highly competitive business landscape, cost optimization is not just a desirable goal; it has become a strategic imperative for sustainable growth and success.

As organizations increasingly migrate their workloads to the cloud, the complexity of managing resources and controlling costs grows exponentially. Without proactive cost optimization strategies, AWS bills can escalate rapidly, leading to budget overruns and reduced financial flexibility. Here we have implemented AWS Instance Scheduler to automate the start and stop times of instances based on workload demands. This approach can significantly reduce costs during non-business hours and idle periods. We have also set up methodology which sets alarms to receive notifications when costs exceed defined thresholds, allowing timely actions to be taken. We have implemented best practices to categorize resources and monitor cost allocation accurately by setting up a common dashboard for the service used by organization which can track and manage cost centers effectively.

In future, this can be extended by including other services like ECS and BEAN Stock.

9. Reference

[1] A. Alalawi, A. M. (2020). A survey for AWS cloud development tools and services. *3rd Smart Cities Symposium (SCS 2020)*, (pp. 17-23). Chicago, USA.

[2] A. Greenberg, J. H. (2008). The cost of a cloud: research problems in data center networks. *ACM SIGCOMM Computer Communication Review*, (pp. 68–73).

[3] A. Sun, G. G. (2018). One Quantifiable Security Evaluation Model for Cloud Computing Platform. *Sixth International Conference on Advanced Cloud and Big Data (CBD)*, (pp. 197-201). Lanzhou, China.

[4] Che, D. a.-C. (2011). A Novel "Credit Union" Model of Cloud Computing. *Digital Information and Communication Technology and Its Applications*, (pp. 1-49). Washington, DC.

[5] Cheriet, A. S. (2014). Cloud Computing: A Risk Assessment Model. *IEEE International Conference on Cloud Engineering*, (pp. 147-152). Boston,USA.

[6] Dutta, T. M. (2014). C-Cloud: A Cost-Efficient Reliable Cloud of Surplus Computing Resources. *IEEE 7th International Conference on Cloud Computing*, (pp. 986-987). Anchorage, USA.

[7] Humphrey, M. M. (2011). Auto-scaling to minimize cost and meet application deadlines in cloud workflows. *2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, (pp. 559–570). Los Angeles.

[8] Ibrahimi, L. N. (2017). Comparison of the cloud computing platforms provided by Amazon and Google. *Computing Conference*, (pp. 236-243). London, UK.

[9] J. F. Salindeho, J. H. (2021). Cost-Benefit Analysis of Cloud Computing in Education Using the Base Cost Estimation Model. *3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, (pp. 1-5). Makasar, Indonesia.

[10] K. Zhang, T. W. (2017). A New Energy Efficient VM Scheduling Algorithm for Cloud Computing Based on Dynamic Programming. *IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, (pp. 249-254). New York, NY, USA.

[11] Lian, Y. C. (2017). Cloud Bursting Scheduler for Cost Efficiency. *IEEE 10th International Conference on Cloud Computing (CLOUD)*, (pp. 774-777). Honololu, USA.

[12] Liang Luo, W. W. (2012). A resource scheduling algorithm of cloud computing based on energy efficient . (pp. 1-6). San Jose, CA: 2012 International Green Computing Conference (IGCC).

[13] Luo L, W. W. (2014). Energy Modeling Based on Cloud Data Center. *Journal of Software*, (pp. 1371-1387). California.

[14] M. Theoharidou, N. T. (2013). In Cloud We Trust: Risk-Assessment-as-a-Service. *7th IFIP WG 11.11 International Conference*, (pp. 100-110).

[15] Mengistu, A. A. (2017). No Data Center" Solution to Cloud Computing. *IEEE 10th International Conference on Cloud Computing (CLOUD)*, (pp. 714-717). Honololu, HI, USA.

[16] Peng, J. Z. (2009). Comparison of several cloud computing platforms. *Second International Symposium on Information Science and Engineering*, (pp. 23-27).

[17] Rahaman, N. I. (2019). A Comparative Study of Major Cloud Service providers for Cloud Computing. *IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, (pp. 100-110). Japan.

[18] Rahaman, N. I. (2019). A Comparative Study of Major Cloud Service providers for Cloud Computing. *IEEE 4th International Conference on Computer and Communication Systems*, (pp. 61–72).

[19] S. Chaisiri, B. -S. (2012). Optimization of Resource Provisioning Cost in Cloud Computing. *IEEE Transactions on Services Computing, vol. 5, no. 2,* (pp. 164-177).

[20] Samimi, P. a. (2011). Review of pricing models for grid & cloud computing. *Computers & Informatics (ISCI)*, (pp. 23-27).