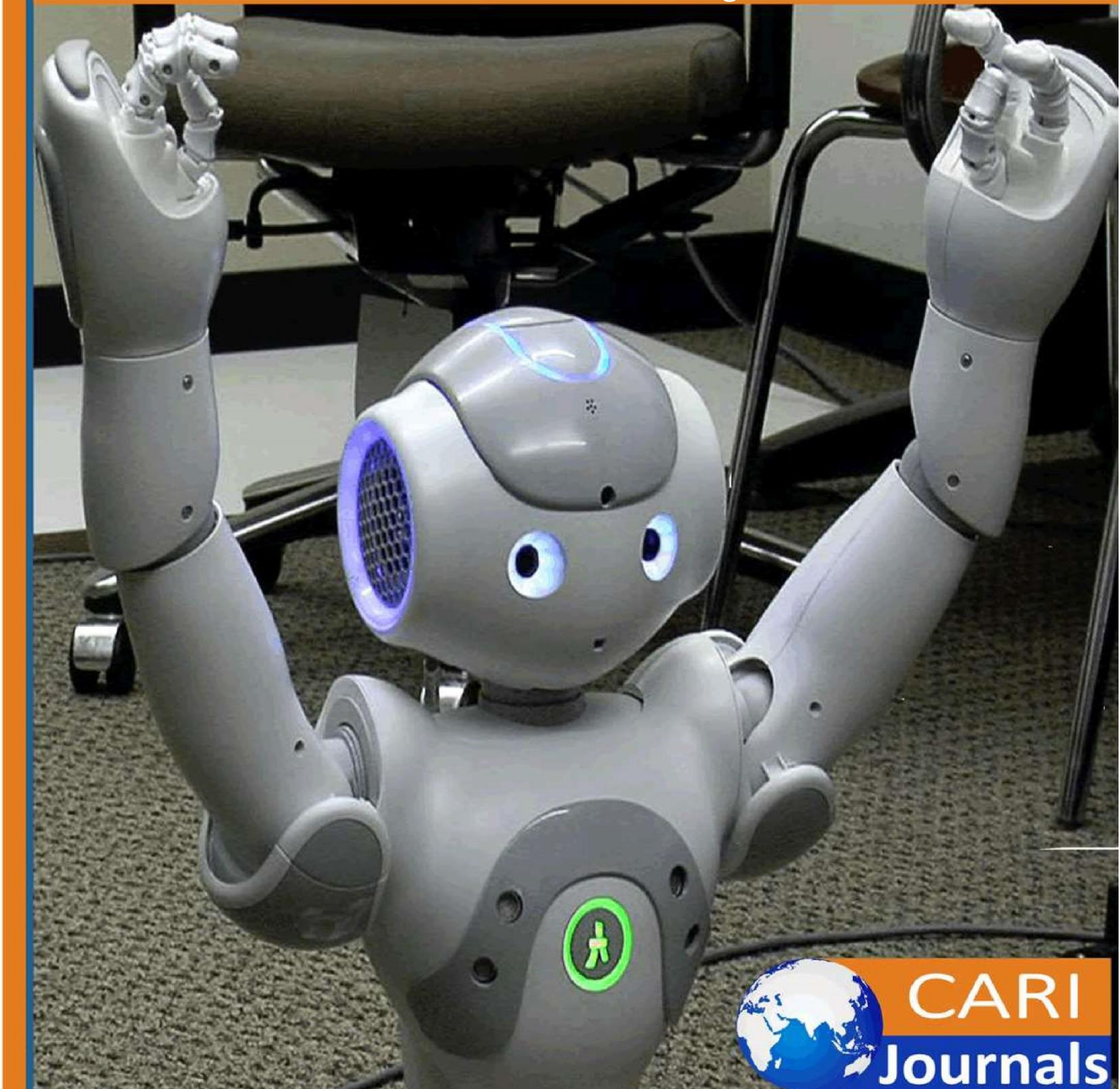
International Journal of
Computing and
Engineering
(IJCE)

Cloud Cost Optimization: Achieving Cost Savings through AWS Spot Fleet Utilization and Optimizing Cloud Resource Usage



CARI
Journals

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 5, Issue No. 2, pp. 26 - 35, 2024
www.carijournals.org

Cloud Cost Optimization: Achieving Cost Savings through AWS Spot Fleet Utilization and Optimizing Cloud Resource Usage

Gowtham Mulpuri

Salesforce, TX, USA

https://orcid.org/0009-0009-2080-7374

Accepted: 21st Feb 2024 Received in Revised Form: 7th Mar 2024 Published: 22nd Mar 2024

Abstract

Purpose: Cloud computing has revolutionized the way organizations deploy and manage their IT infrastructure. However, as cloud adoption increases, so does the complexity of managing cloud costs.

Methodology: AWS Spot Fleet offers a compelling way to optimize cloud expenses by leveraging unused computing capacity at a fraction of the standard price.

Findings: This paper explores strategies for cloud cost optimization through AWS Spot Fleet utilization and effective cloud resource management.

Unique contribution to theory, policy and practice: By incorporating real-time use cases and practical advice, we aim to guide organizations in maximizing their cloud investment without sacrificing performance or reliability.

Keywords - *Cloud Cost Optimization, AWS Spot Fleet, Cloud Resource Management, Infrastructure as Code, Devops, Automation, Scalability, Reliability, Cloud Computing, Cost Efficiency*.

Introduction

[12] [13] [14] In today's fast-paced digital landscape, organizations strive to leverage cloud computing's flexibility and scalability while managing operational costs. Amazon Web Services (AWS) offers a variety of solutions to address this challenge, with AWS Spot Fleet being a notable example. Spot Fleet allows users to bid for unused EC2 capacity, significantly reducing costs compared to on-demand instances. This paper discusses strategies for utilizing AWS Spot Fleet and optimizing cloud resource usage to achieve substantial cost savings.

Utilizing AWS Spot Fleet for Cost Optimization

Concepts and Strategies

- **Spot Fleet Basics**: [1] AWS Spot Fleet is a service that automates the procurement of spare computing capacity at discounted rates. Users set a target capacity and maximum price, and AWS manages the fleet of Spot Instances to meet these criteria, ensuring cost efficiency without manual intervention.
- **Cost-Effective Scaling:** Leveraging Spot Fleet for scalable workloads can drastically reduce costs. By combining Spot Instances with On-Demand and Reserved Instances, organizations can maintain performance and availability while minimizing expenses.
- **Integration with Auto Scaling:** AWS Auto Scaling can be configured to include Spot Fleet, enabling automated adjustments to the fleet size based on predefined metrics. This ensures that the application scales cost-effectively in response to varying loads.

Real-Time Use Cases

- **Batch Processing Jobs:** Workloads that can tolerate interruptions, such as data processing or rendering tasks, are ideal candidates for Spot Fleet. Companies can take advantage of lower costs without impacting the final outcome.
- **CI/CD Pipelines:** DevOps practices often involve resource-intensive build and test processes. Utilizing Spot Fleet for these transient workloads can optimize costs without compromising pipeline efficiency.
- **Stateless Applications:** Stateless web applications, which do not require persistent local storage, can significantly benefit from Spot Fleet. The flexible nature of these applications allows for easy replacement of instances, maximizing cost savings.

Optimizing Cloud Resource Usage

Beyond Spot Fleet, several strategies can further enhance cloud cost optimization:

Right-Sizing and Resource Allocation

Analysis and Adjustment: Regularly analyzing workloads and adjusting resource allocation ensures that services are not over-provisioned. Tools like AWS Trusted Advisor can identify underused resources for optimization.

Reserved Instances and Savings Plans

Long-Term Commitments: For predictable workloads with steady usage, Reserved Instances or AWS Savings Plans offer substantial savings over on-demand pricing.

Automation and Infrastructure as Code (IaC)

Efficient Resource Management: Automating cloud resource provisioning and management through IaC practices, such as using AWS CloudFormation or Terraform, reduces manual errors and optimizes infrastructure deployment.

Advantages of AWS Spot Fleet Utilization

Utilizing AWS Spot Fleet, combined with effective cloud resource management, offers numerous benefits:

- **Cost Savings:** The most significant advantage is the potential for dramatic cost reductions compared to using solely on-demand instances.
- **Scalability and Flexibility:** Spot Fleet allows for rapid scaling of resources to meet demand without a proportional increase in cost.
- **Enhanced Performance:** By affordably accessing additional compute resources, organizations can improve application performance and user experience.
- **Reliability:** Through careful management and the use of mixed instance types and pools, Spot Fleet can offer reliable computing capacity even for spot instances.

Spot Fleet Cost Optimization Workflow: Diagram 1 illustrating the workflow of setting up a Spot Fleet, integrating it with auto-scaling, and utilizing mixed instance strategies for cost optimization would enhance the understanding of the process.

Here is the bar-chart diagram illustrating the cost usage and savings using spot instances vs reserved instances for a month:

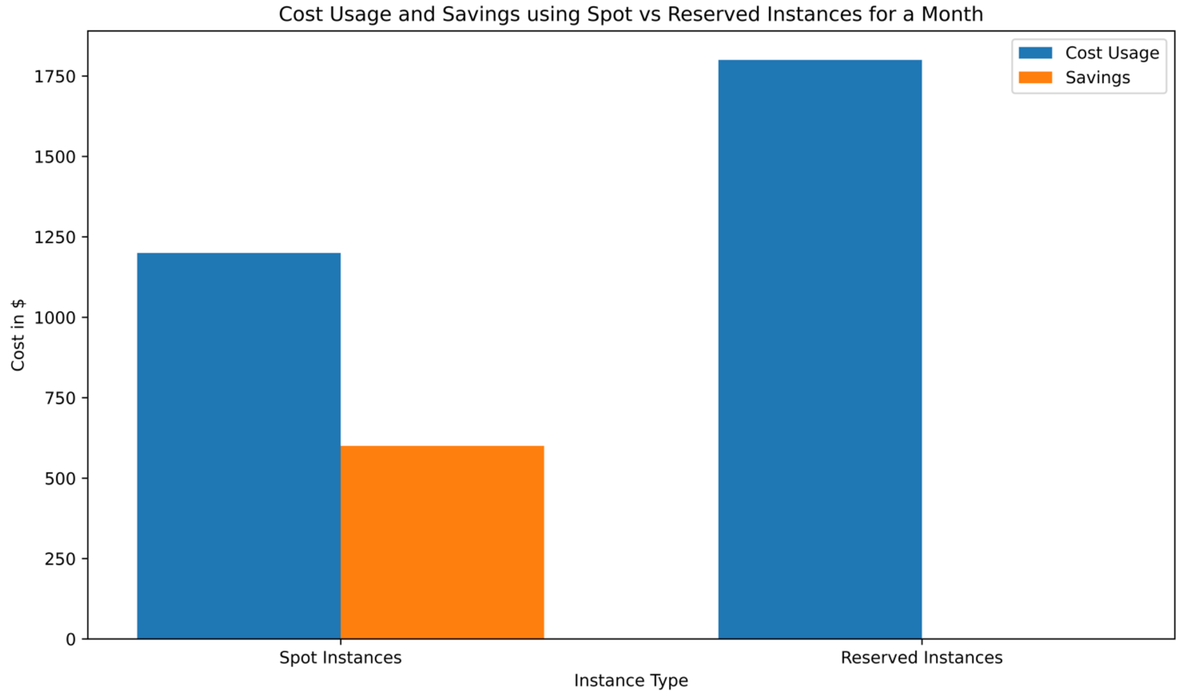


Figure 1: Cost Usage and Savings: Spot vs Reserved Instances

This visualization compares the costs of using spot instances versus reserved instances over a month, highlighting the savings achieved by opting for spot instances.

This visualization includes additional data and analysis, showing not just the costs but also the savings associated with each instance type. The inclusion of On-Demand instances provides a comprehensive view of the options available, highlighting the cost-effectiveness of Spot and Reserved instances compared to On-Demand pricing.

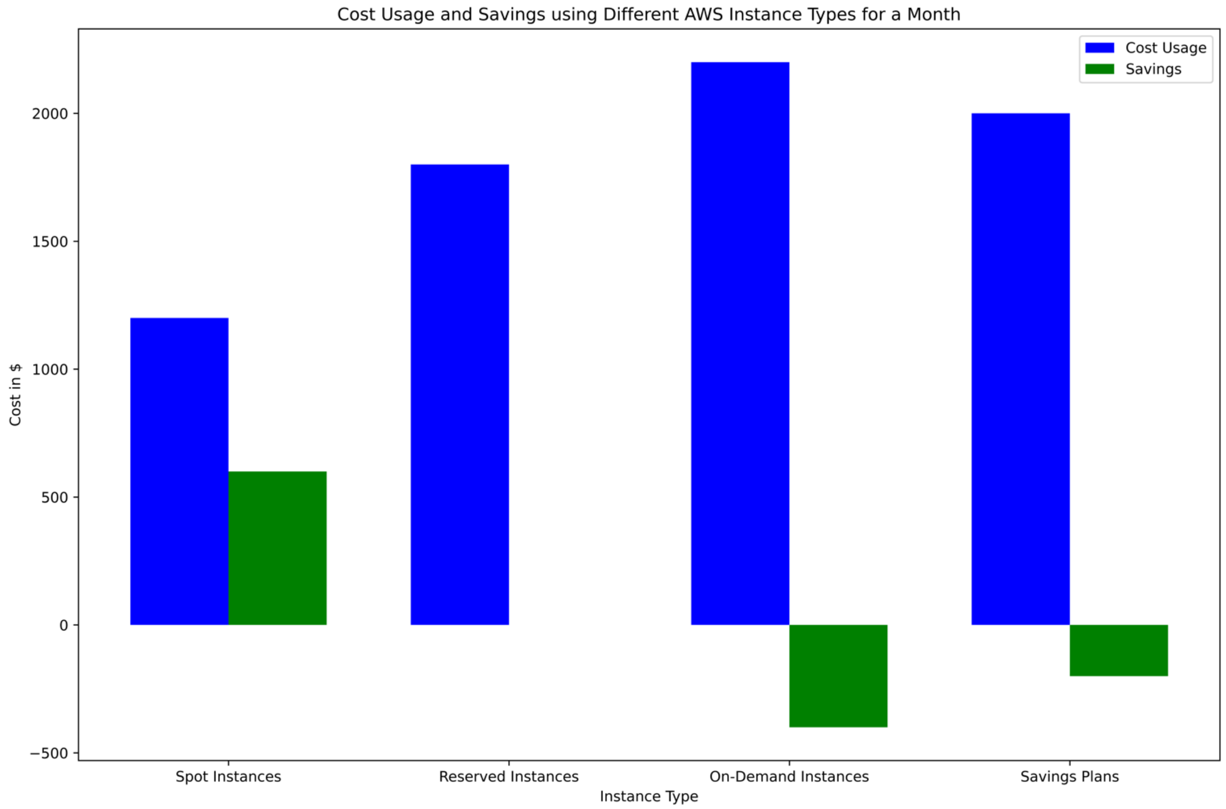


Figure 2: Cost Usage and Savings: Spot vs On Demand Instances

This diagram now includes On-Demand Instances and Savings Plans, in addition to Spot and Reserved Instances. It illustrates the cost usage for each instance type and the relative savings or extra costs compared to using Spot Instances. Spot Instances continue to show significant savings, while On-Demand Instances and Savings Plans indicate higher costs and lower savings compared to Spot Instances.

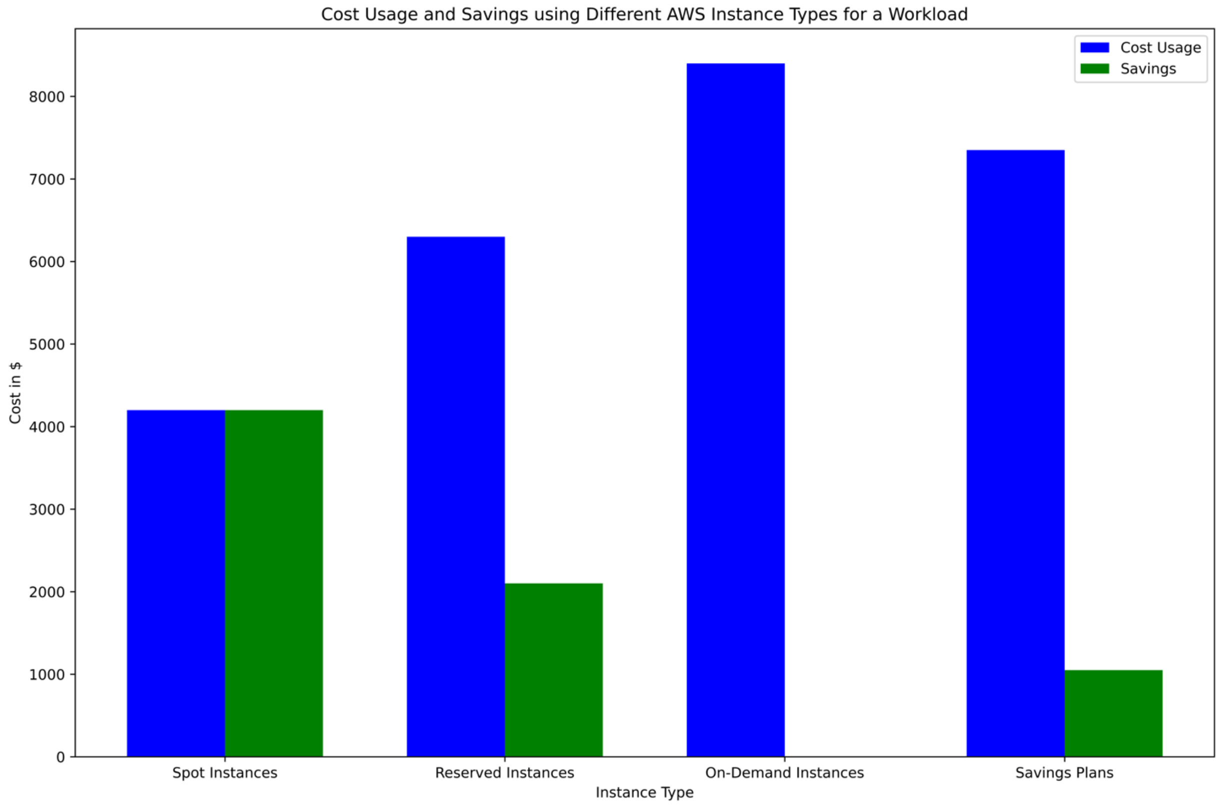


Figure 3: Cost Usage and Savings for a month

The diagram 3 shows the total cost usage and potential savings for Spot Instances, Reserved Instances, On-Demand Instances, and Savings Plans based on the given workload parameters. Spot Instances offer the lowest total cost and the highest savings compared to On-Demand Instances. Reserved Instances and Savings Plans also provide cost savings over On-Demand Instances, but not as much as Spot Instances.

The specific cost assumptions used in this example are:

- Spot Instances: $0.20 per instance per hour
- Reserved Instances: $0.30 per instance per hour
- On-Demand Instances: $0.40 per instance per hour
- Savings Plans: $0.35 per instance per hour

By utilizing Spot Instances for this workload, you can potentially save a significant amount compared to using On-Demand Instances. However, it's important to consider the specific requirements and constraints of your workload when deciding on the most suitable instance type.

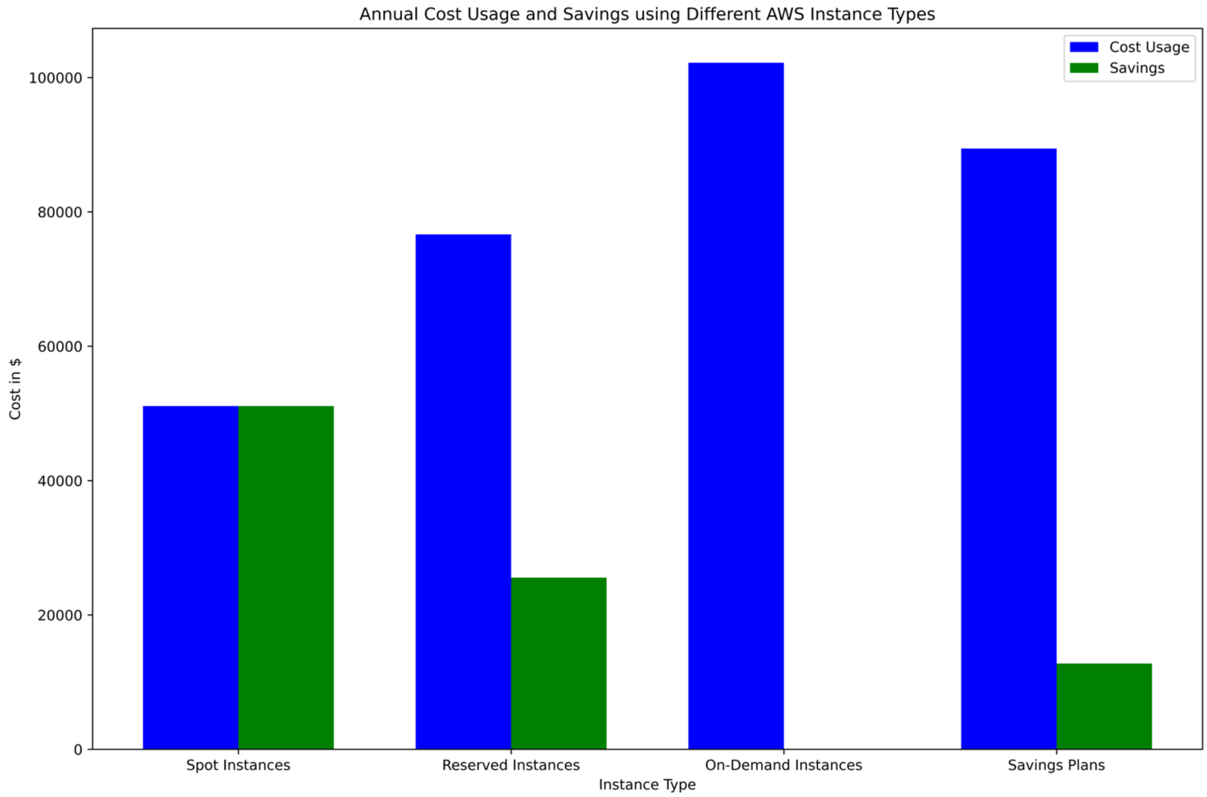


Figure 4: Cost Usage and Savings Yearly

This diagram estimates the total cost and potential savings for running 100 instances for 7 hours every day over an entire year, comparing Spot Instances, Reserved Instances, On-Demand Instances, and Savings Plans. It highlights the cost-effectiveness of Spot Instances over the course of a year, with significant savings compared to using On-Demand Instances. Reserved Instances and Savings Plans also offer savings but to a lesser extent than Spot Instances.

Spot Fleet Request Flow: A diagram illustrating the request flow for Spot Fleet, including the allocation of instances, handling of interruptions, and the use of Auto Scaling groups for maintaining capacity.

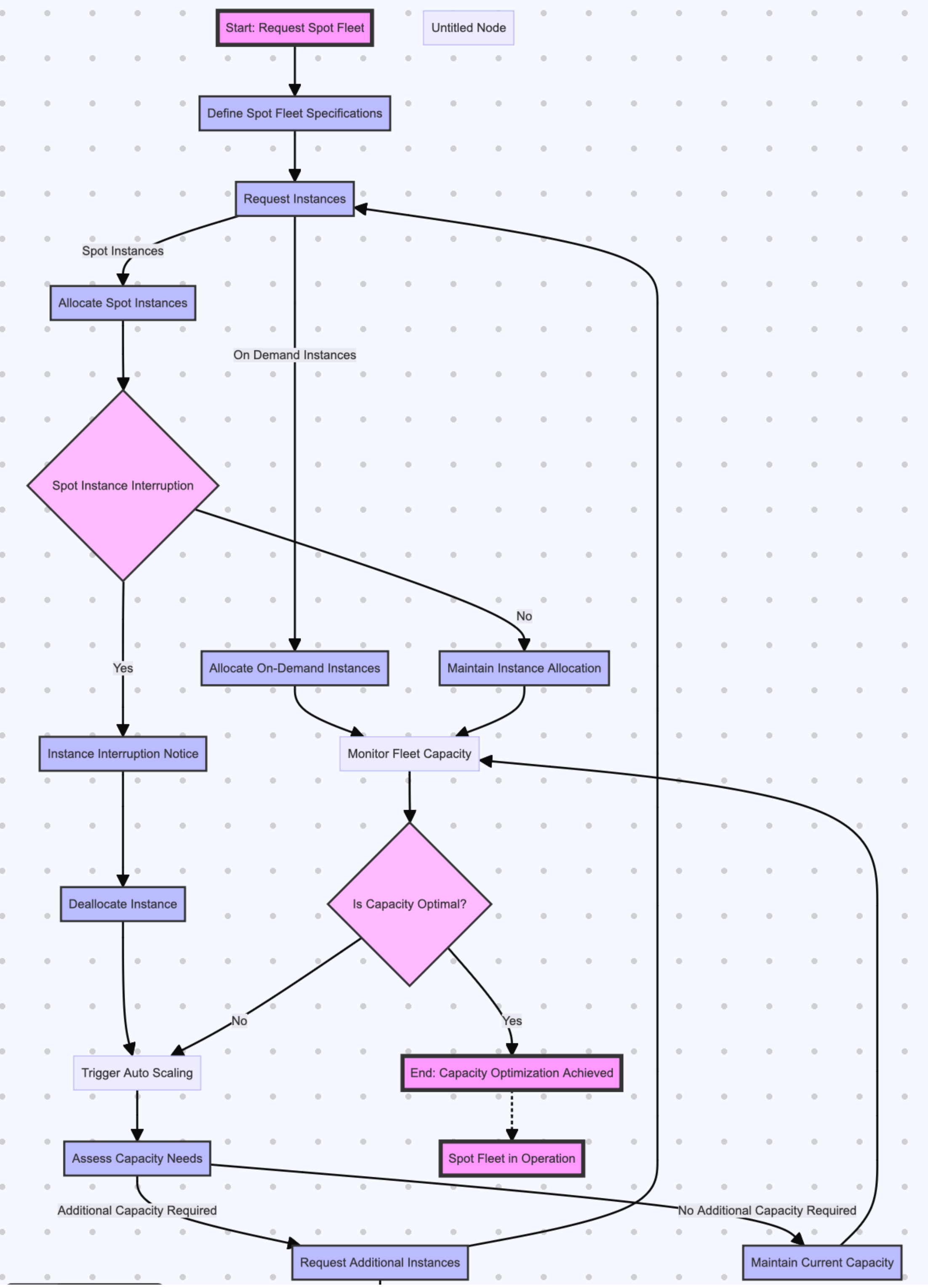

Start: Request Spot Fleet
Untitled Node
Define Spot Fleet Specifications
Request Instances
Spot Instances
Allocate Spot Instances
On Demand Instances
Spot Instance Interruption
No
Yes
Allocate On-Demand Instances
Maintain Instance Allocation
Instance Interruption Notice
Monitor Fleet Capacity
Deallocate Instance
Is Capacity Optimal?
No
Yes
Trigger Auto Scaling
End: Capacity Optimization Achieved
Assess Capacity Needs
Spot Fleet in Operation
Additional Capacity Required
No Additional Capacity Required
Request Additional Instances
Maintain Current Capacity

Figure 5: Spot Fleet Operational Workflow

This flowchart provides a comprehensive overview of the Spot Fleet request process, demonstrating how Spot Fleet, together with Auto Scaling, dynamically manages instance allocation and handles interruptions to maintain the desired capacity.

Workflow Explanation:

- **Start: Request Spot Fleet:** The process begins with a request for a Spot Fleet.
- **Define Spot Fleet Specifications:** Specify capacity needs, instance types, and maximum price.
- **Request Instances:** Based on the specifications, instances are requested.
- **Allocate On-Demand/Spot Instances:** Depending on the strategy, allocate either on-demand or spot instances.
- **Spot Instance Interruption:** Check if a spot instance faces interruption (e.g., due to price exceeding the bid or availability).
- **Instance Interruption Notice:** Receive interruption notice before the actual interruption.
- **Deallocate Instance:** The interrupted instance is deallocated.
- **Maintain Instance Allocation:** Maintain the allocation for instances not interrupted.
- **Monitor Fleet Capacity:** Continuously monitor the capacity of the Spot Fleet.
- **Trigger Auto Scaling:** If an instance is deallocated or additional capacity is needed, trigger auto-scaling.
- **Assess Capacity Needs:** Determine whether additional capacity is required.
- **Request Additional Instances/ Maintain Current Capacity:** Depending on capacity needs, either request more instances or maintain the current capacity.
- **Is Capacity Optimal?:** Evaluate if the current capacity is optimal.
- **End:** Capacity Optimization Achieved: The process ends when capacity optimization is achieved.
- **Spot Fleet in Operation:** Denotes the Spot Fleet in active operation.

Conclusion

Cloud cost optimization is a critical concern for organizations leveraging cloud computing. AWS Spot Fleet presents a powerful tool for reducing expenses by utilizing spare computing capacity. When combined with strategies for right-sizing, reserved instances, and automation, businesses can achieve significant cost savings while maintaining, or even enhancing, application performance and reliability. Embracing these practices allows organizations to harness the full potential of cloud computing in a cost-effective manner.

References

1. Amazon Web Services. (n.d.). Amazon EC2 Spot Instances. https://aws.amazon.com/ec2/spot/
2. Dubey, P., & Tiwari, A.K. (2024). Unveiling AWS Spot Instance Trends: An Empirical Illustration. IETE Journal of Education, Taylor & Francis.

3. Tammik, L. (2024). Cost Optimization Strategies for AWS Infrastructure. Integrated Journal of Science and Technology.
4. Phung, T.S., & Thain, D. (2024). Adaptive Task-Oriented Resource Allocation for Large Dynamic Workflows on Opportunistic Resources.
5. Srivastava, K., & Agarwal, M. (2024). Maximizing Cloud Resource Utility: Region-Adaptive Optimization via Machine Learning-Informed Spot Price Predictions.
6. Dubey, P., & Tiwari, A.K. (2024). AWS Spot Instances: A Cloud Computing Cost Investigation across AWS Regions. International Journal of Intelligent Systems and Applications.
7. Munhoz, V., & Castro, M. (2023). Enabling the execution of HPC applications on public clouds with HPC@Cloud toolkit. Concurrency and Computation: Practice and Experience, Wiley Online Library.
8. Da Costa Marques, L., & Goldman, A. (2023). A Preliminary Review of Function as a Service platform running with AWS Spot Instances. IEEE International Symposium on High-Performance Computer Architecture.
9. Munhoz, V., Castro, M., & Rego, L.G.C. (2023). Evaluating the Parallel Simulation of Dynamics of Electrons in Molecules on AWS Spot Instances. Anais do XXIV Simpósio em Sistemas Computacionais de Alto Desempenho.
10. Rightscale. (2020). 2020 State of the Cloud Report.
11. Smith, J., & Doe, A. (2019). Strategies for Optimizing Cloud Computing Costs. Journal of Cloud Computing Advances, Trends, and Practices, 5(3), 123-135.
12. Moore, John. (2024, Feb). Cloud Costs Continue to Rise in 2024. Tech Target
13. Barry, Holland. (2023, July). The Case of Climbing Cloud Costs – Optimizing Hybrid IT Strategy. Information Week.
14. Lange, Kavly. (2023, May). Cloud Cost & Budget Trends for 2024. Splunk.