

Question 1: Imagine you're working with a client who's concerned about managing their AWS costs efficiently. What strategies or recommendations would you propose to help them reduce their AWS costs?

Answer:

Here are some strategies that you can use to reduce your AWS costs:

1. **Right Sizing:** Ensure that you're using the right instance types and sizes for your application needs. For example, if you're running a memory-intensive application, choose an instance type optimized for memory. If your instance is consistently at low CPU utilization, you might be over-provisioned. You could save costs by downsizing to a smaller instance type.
 - *Cost Calculation:* If an m5.2xlarge instance costs \$0.404 per hour and you're only using 25% of its capacity, you could potentially save about 75% by downsizing to an m5.large instance which costs \$0.101 per hour.
 - ◆ *Cost of m5.2xlarge per month:* 1 instances x \$0.404 On Demand hourly cost x 730 hours in a month = \$294.920000
 - ◆ *Cost of m5.large per month:* 1 instances x 0.101 On Demand hourly cost x 730 hours in a month = \$73.730000
 - *Savings:* \$294.92 – \$73.73 = **\$221.19** saved per month!!!
2. **Reserved Instances:** If you have steady state usage, Reserved Instances (RI) can provide significant savings compared to On-Demand instance pricing. For example, a 1-year reserved m5.2xlarge instance costs \$0.192 per hour, which is 50% cheaper than the on-demand price.
 - *Cost Calculation:* If you're running an m5.2xlarge instance 24/7 for a year, the total cost on-demand vs reserved:
 - ◆ *Cost of on-demand m5.2xlarge instance:* 1 instances x \$0.404 On Demand hourly cost x 730 hours in a month x 12 months = \$3,539.04
 - ◆ *Cost of reserved m5.2xlarge instance:* 1 instances x \$0.255 x 730 hours in a month x 12 months = \$2,233.80
 - *Savings:* \$3,539 - \$2,233 = **\$1,306** saved per month!!!
3. **Spot Instances:** If your applications are fault-tolerant and can withstand interruptions, Spot Instances can save you up to 90% compared to On-Demand pricing. For example, a spot m5.2xlarge instance can cost as low as \$0.0384 per hour.
 - *Cost Calculation:* If you're running an m5.2xlarge instance 24/7 for a month, the total cost on-demand vs spot instance:

- ◆ Cost of m5.2xlarge per month: 1 instances x \$0.404 On Demand hourly cost x 730 hours in a month = \$294.92
 - ◆ 1 instances x \$0.404 On Demand hourly cost x 730 hours in a month 29 x 0.1 = \$29.492
 - *Savings:* \$294.92 - \$29.492 = **\$265.428** saved per month!!!
- 4. **Delete Unused Resources:** Regularly review and delete unused resources such as unattached EBS volumes, old snapshots, and idle load balancers.
 - *Cost Saving Calculation:* If you have 10 unattached EBS volumes costing \$0.10 per GB-month of provisioned storage, and each volume is 100GB, the total cost would be 10 x \$0.10 x 100 = \$100 per month. By deleting these unused volumes,
- 5. **Optimize Snapshots:** While snapshots are a great way to back up your data, they can accumulate over time and lead to higher costs. Consider implementing a snapshot lifecycle policy that automatically deletes older snapshots.
 - *Cost Saving Calculation:* Suppose you have 50 snapshots, each with a storage of 100GB. If the cost per GB-month of snapshot storage is \$0.05.
 - ◆ The total cost of these snapshots would be 50 snapshots x 100GB/snapshot x \$0.05/GB-month = \$250/month.
 - ◆ If you implement a lifecycle policy that reduces the number of snapshots by half, you could save **\$125** per month!
 - ◆ You could save **\$100** per month!
- 6. **Serverless Architectures:** By using serverless architectures like AWS Lambda, you only pay for the compute time you consume. There is no charge when your code is not running. You can run code for virtually any type of application or backend service with zero administration.
 - **Cost Saving Calculation:**
 - ◆ Suppose you have an application that runs on an EC2 instance 24/7, costing \$0.02 per hour (or \$14.40 per month).
 - ◆ If you were to move this application to Lambda and it runs for 1.5 million seconds per month (or about 416 hours), at a rate of \$0.0000166667 per second, the total cost would be \$25.
 - ◆ If the application is idle 50% of the time, you could save \$14.40 - \$12.50 = \$1.90 per month by moving to a serverless architecture.
- 7. **Use Cost Allocation Tags:** Implement a strong tagging strategy to understand who is spending what and why. This can help you identify cost-saving opportunities. This is more of a cost management strategy rather than a direct cost-saving measure. It can help you identify areas where you can save costs.