

The background of the slide features a blurred image of an Airbnb office interior. On the left, a large, illuminated Airbnb logo is visible on a wall. In the center, a person is seen from behind, working at a desk with multiple computer monitors. The overall lighting is dim, with the primary light source being the glowing logo and office lights in the distance.

# Airbnb's Cost Optimization Strategies on AWS

ADOPTION AND MIGRATION PLANNING – SYST8171

**PRESENTED TO**  
VIGNESH RAVI

**PRESENTED BY**  
SWATHI ANIL(8905477)

# Optimization Strategies:

## Strategies Used by Airbnb

- ✓ Savings Plan: Airbnb changed its approach to using EC2 from on-demand instances to using a Saving Plan. This plan provides discounts for dedicated use over some time of one to three-year term, which helps in cost reduction.
- ✓ S3 Intelligent-Tiering: Airbnb makes use of S3 Intelligent-Tiering, an automated storage class that helps in moving objects based on the patterns of usage.
- ✓ S3 Glacier Storage: Airbnb has the S3 Glacier storage class for storing backups and other data that is rarely accessed. This storage class is cheaper compared to the on-demand storage classes.
- ✓ Right-Sizing with Auto Scaling: Airbnb, in association with the use of Amazon EC2, may implement characteristics of auto-scaling. This enables them to acquire the right equipment to facilitate their high workload during busy hours, without incurring high costs.
- ✓ AWS Lambda: For serverless computing requirements, they use AWS Lambda. This enables them to execute applications without worrying about servers, thus saving much on infrastructure and the process.
- ✓ Spot Instances: Some of the least critical applications that can tolerate interruption may be economically profited to run on AWS Spot instances by Airbnb.





# Suggested Improvements

## Cloud Cost Optimization Tools.

- The cloud cost optimization tool can also be integrated for even a better management of cost aspect for Airbnb. The use of these tools offer full information concerning cost and resources that are underutilized and areas of improvements.

## Automated Scaling:

- Automate scaling of the resources is an effective way of avoiding situations whereby resources are used sparingly, yet money is spent on resources that are hardly used. Some tools can even have the functionality of auto-scaling the compute instances on the basis of real-time requirements.

## Reserved Instances for Predictable Workloads:

- Thus, while with Savings Plans the provision of flexibility is given, reserved instances may be even cheaper for the really consistent workloads. These provide capacity commitment at a certain price that is lower than the spot price.

## Containerization with Cost-Optimized Platforms:

- Docker or Kubernetes like technologies for the containerization of microservices architectures.
- It can be seen that containerization also aids in the separation of the application and their respective resources while providing efficient resource management.



# CONCLUSION

Specifically, cloud cost optimization can be seen in the use of AWS Cost & Usage Report, AWS Savings Plans, S3 Storage classes, cost allocation tags used in Airbnb. These strategies have inevitably led to the reduction of costs and better usage of resources.

Introducing the suggested changes such as the AI-based forecast, the meaningful utilization of Spot Instances, a more robust cost control system, and employees' training, Airbnb may reach an even greater level of cost-effectiveness and a long-term perspective of a company's cloud usage.

# References

1. *Optimizing Usage and Costs by Using Savings Plans and Actionable Cost Data on AWS | Airbnb Case Study | AWS*. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/solutions/case-studies/airbnb-optimizes-usage-and-costs-case-study/>
2. *5 Effective EC2 Cost Optimization Strategies to Maximize Savings | CloudKeeper*. (n.d.). Wwww.cloudkeeper.com. Retrieved July 24, 2024, from <https://www.cloudkeeper.com/insights/blog/5-effective-ec2-cost-optimization-strategies-maximize-savings>

