**Strategy for Optimizing Resource Usage and Costs Without Compromising Security and Scalability**

**1. Right-Sizing Resources**

* **EC2 Instances**: Regularly monitor instance performance and usage metrics using Amazon CloudWatch to ensure they are appropriately sized for the workload. Choose instance types that provide the best balance between performance and cost.
* **Auto-Scaling**: Implement auto-scaling groups for both frontend and backend EC2 instances. This ensures that the number of running instances dynamically adjusts based on demand, minimizing costs during low-traffic periods while maintaining performance during peak loads.

**2. Optimized Storage Solutions**

* **Amazon S3**: Utilize S3 lifecycle policies to transition infrequently accessed data to cheaper storage classes such as S3 Infrequent Access or Glacier, reducing storage costs without sacrificing data availability.
* **Amazon RDS**: Use automated backups and snapshots with careful retention policies to avoid excessive storage costs. Consider using RDS reserved instances for predictable, long-term database workloads, which can provide significant cost savings over on-demand instances.

**3. Efficient Database Management**

* **RDS Read Replicas**: Deploy RDS read replicas to handle read-heavy workloads, distributing the read traffic and reducing the load on the primary database instance. This enhances both performance and scalability without significantly increasing costs.
* **ElastiCache**: Implement ElastiCache for caching frequently accessed data, which reduces the load on the RDS instance and improves application response times. By offloading frequent queries to ElastiCache, you can scale the database layer more cost-effectively.

**4. Network Optimization**

* **Single NAT Gateway**: Use a single NAT gateway for the VPC, strategically placed in a public subnet to provide outbound internet access for resources in private subnets. This minimizes the number of NAT gateways required, reducing costs while maintaining necessary functionality.
* **Efficient Routing**: Configure route tables to ensure efficient traffic flow between subnets, reducing latency and improving overall network performance. Use VPC peering or AWS Transit Gateway to optimize inter-VPC traffic if needed.

**5. Security Best Practices**

* **Least Privilege Principle**: Implement IAM policies that follow the principle of least privilege, granting users and services only the permissions they need to perform their tasks. This minimizes the risk of unauthorized access and potential security breaches.
* **Security Groups and Network ACLs**: Regularly review and update security group rules to ensure they are not overly permissive. Implement network ACLs for an additional layer of security, restricting traffic at the subnet level.
* **AWS WAF**: Use AWS WAF to protect against common web exploits and attacks, ensuring that only legitimate traffic reaches your application. This helps maintain security without requiring additional infrastructure costs.

**6. Cost Management and Monitoring**

* **AWS Cost Management Tools**: Utilize AWS Cost Explorer and AWS Budgets to track and manage costs. Set up alerts for unusual spending patterns to promptly address any cost anomalies.
* **Monitoring and Logging**: Implement comprehensive monitoring using AWS CloudWatch and logging using AWS CloudTrail. Optimize log retention policies to balance between keeping essential logs for troubleshooting and compliance, and avoiding excessive storage costs.

**7. Serverless and Managed Services**

* **AWS Lambda**: For event-driven and short-duration tasks, consider using AWS Lambda. This can significantly reduce costs as you only pay for the compute time you consume, and it automatically scales with demand.
* **Managed Services**: Leverage managed services like Amazon RDS, ElastiCache, and S3, which offload maintenance tasks and allow you to focus on your application. Managed services are typically more cost-effective and provide built-in scalability and security features.

**8. Regular Audits and Reviews**

* **Resource Utilization Audits**: Conduct regular audits of your AWS environment to identify underutilized resources. Decommission or downsize resources that are not fully utilized.
* **Cost Optimization Reviews**: Periodically review AWS Trusted Advisor recommendations for cost optimization, performance, security, and fault tolerance. Implement suggested changes to continuously optimize your environment.

By following these strategies, you can effectively optimize resource usage and costs while maintaining high levels of security and scalability for your AWS infrastructure.