Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2020, 7(9):204-208**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Application Stack Migration: Migrating Application Stacks, Such as Moving from Classic ELB to ALB

Gowtham Mulpuri

Silicon Labs, TX, USA
Email: gowtham.mulpuri@silabs.com

Abstract In the dynamic realm of cloud computing, optimizing application delivery is paramount for scalability, performance, and cost-efficiency. Elastic Load Balancers (ELBs) play a critical role in distributing incoming application traffic across multiple targets, ensuring reliability and availability. Amazon Web Services (AWS) offers two predominant types of load balancers: the Classic Load Balancer (CLB) and the Application Load Balancer (ALB). This paper explores the migration process from CLB to ALB, emphasizing the strategic advantages, challenges, and real-world applications of such a transition. Drawing from extensive experience in DevOps and cloud infrastructure management, this analysis provides insights into leveraging ALB features for modern application architectures

Keywords Application Stack Migration, Classic ELB, Application Load Balancer, AWS, Cloud Computing, Scalability, Performance

1. Introduction

The evolution of cloud services has introduced a plethora of tools and services designed to enhance application delivery and performance. Among these, load balancing stands out as a critical component, enabling the distribution of traffic across multiple computing resources to optimize resource use, maximize throughput, minimize response time, and ensure fault tolerance. Historically, AWS users have relied on Classic Load Balancers for their simplicity and broad application. However, the advent of Application Load Balancers offers advanced features tailored for modern application architectures, prompting many organizations to consider migrating their application stacks from CLB to ALB

Migrating from Classic ELB to ALB: Concepts and Considerations

The Classic Load Balancer, while effective in distributing traffic across EC2 instances, operates at layer 4 of the OSI model, offering basic load balancing across multiple EC2 instances. In contrast, the Application Load Balancer operates at layer 7, providing advanced routing, health checks, and features specifically designed for HTTP and HTTPS applications

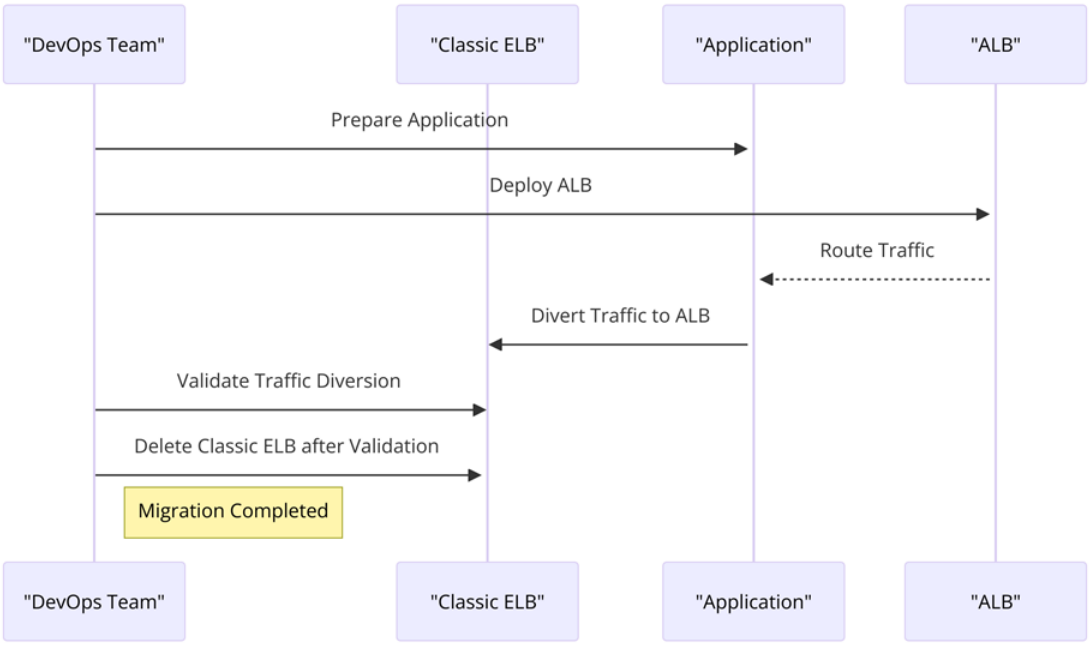

Figure 1: ELB to ALB Migration Concepts

Let's dive into the details of each diagram, focusing on the flow, steps, and methods involved in real time applications:

Detailed Steps for Application Stack Migration

The ***Figure 2*** illustrates the sequence of actions for migrating application stacks from a Classic ELB (Elastic Load Balancer) to an ALB (Application Load Balancer).

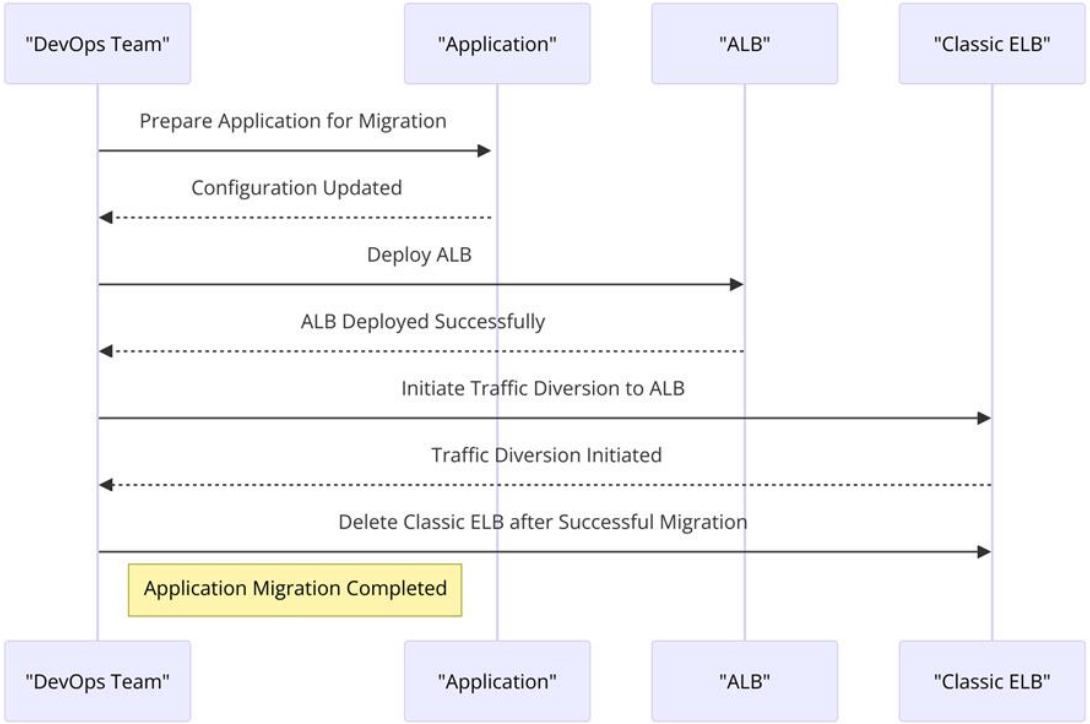

Figure 2: ELB to ALB Migration Concepts

1. **Preparation:** The DevOps team initiates the process by preparing the application for migration. This involves updating configurations to ensure compatibility with ALB and may include adjustments to the application's codebase or deployment settings.
2. **ALB Deployment:** Once the application is ready, the team proceeds to deploy an ALB. This step involves creating and configuring the ALB within the AWS environment to handle incoming traffic according to the predefined rules
3. **Traffic Diversion:** With the ALB in place, traffic originally directed to the Classic ELB begins to divert to the ALB. This step is crucial for a seamless transition, ensuring that the ALB takes over traffic handling without disrupting the user experience.
4. **Validation and Cleanup:** After successfully diverting traffic, the DevOps team validates the ALB's performance and functionality. Upon confirmation, the Classic ELB is decommissioned, completing the migration process.

Error Handling During Migration

The ***Figure 2*** focuses on error handling during the migration process:

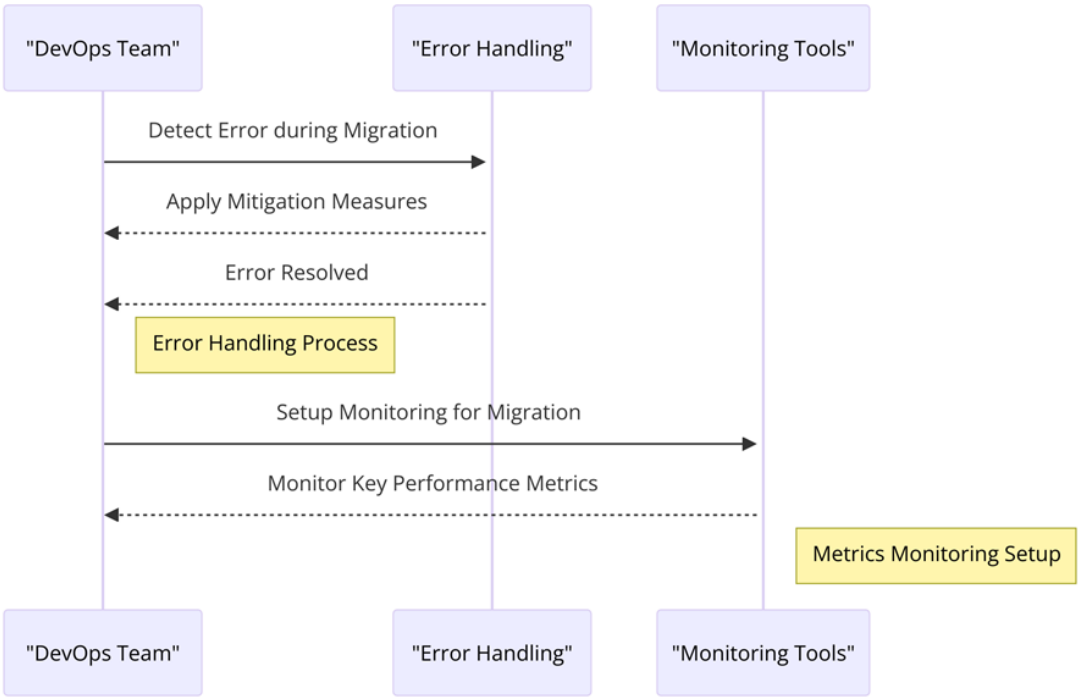

Figure 3: ELB to ALB Migration Error Handling

1. **Error Detection:** The DevOps team monitors the migration for potential errors. This proactive approach ensures that any issues are identified early in the process.
2. **Mitigation Measures:** Upon detecting an error, the team engages error-handling procedures. These may include rollback mechanisms, applying hotfixes, or adjusting configurations to address the issue.
3. **Resolution and Continuation:** Once the error is resolved, the migration process continues. This step emphasizes the importance of having a robust error-handling framework to minimize downtime and ensure a smooth transition.

Metrics Monitoring for ALB Deployment

The fourth diagram highlights the importance of metrics monitoring during the ALB deployment:

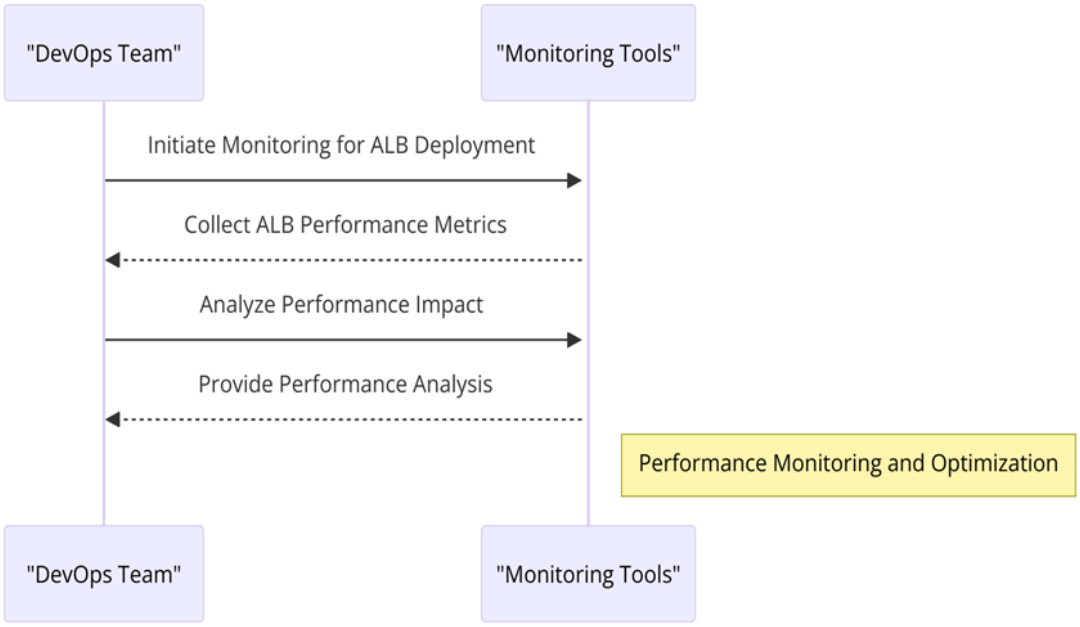

Figure 4: ELB to ALB Migration Metrics Monitoring

1. **Monitoring Setup:** The DevOps team sets up monitoring tools to collect and analyze performance metrics from the ALB. This setup is crucial for gaining insights into the ALB's impact on application performance and availability.
2. **Data Collection:** Monitoring tools collect data on key performance indicators, such as response times, throughput, and error rates. This data is instrumental in assessing the ALB's effectiveness and identifying potential bottlenecks.
3. **Performance Analysis:** The team analyzes the collected metrics to evaluate the ALB's performance. Based on this analysis, optimization strategies may be implemented to enhance efficiency and ensure that the application meets performance expectations post-migration.

In each diagram, the sequence of steps and the methodologies employed reflect a systematic approach to DevOps practices, emphasizing preparation, automation, monitoring, and continuous improvement. The use of ALB over Classic ELB represents a strategic move towards more flexible, efficient, and scalable cloud infrastructure management, aligning with the principles of DevOps to support rapid deployment and high availability.

Key considerations for migration include:

- **Compatibility and Requirements:** Assessing application compatibility with ALB, particularly for applications that use HTTP/HTTPS protocols.
- **Feature Set:** Understanding the advanced features of ALB, such as content-based routing, WebSocket support, and containerized application support.
- **Cost Implications:** Evaluating the cost differences between CLB and ALB, as ALB may offer cost savings for certain traffic patterns and configurations.

Advantages of Migrating to ALB

- **Advanced Routing Capabilities:** ALB supports content-based routing, allowing requests to be routed based on the content of the request itself. This is particularly useful for microservices architectures and containerized applications, where different paths within the same application may require different backend services.
- **Improved Performance:** ALB offers enhanced performance for modern web applications, including support for HTTP/2 and WebSocket, which can reduce load times and improve the user experience.

- **Enhanced Monitoring and Logging:** With ALB, organizations gain access to detailed access logs and metrics, enabling more precise monitoring and troubleshooting of application performance issues.
- **Security and Compliance:** ALB supports advanced security features such as SSL/TLS offloading, which can simplify compliance with security standards.

Real-Time Use Cases

1. **Microservices Architecture:** A financial services company migrated its monolithic application to a microservices architecture, leveraging ALB's content-based routing to direct traffic to the appropriate service based on the request path.
2. **High-Performance Web Applications:** An e-commerce platform experiencing slow load times and performance issues migrated to ALB to take advantage of HTTP/2 support, significantly improving page load speeds and customer satisfaction.
3. **Containerized Workloads:** A software company running containerized workloads on Amazon ECS benefited from ALB's integration with ECS, enabling dynamic port mapping and simplified service discovery.

Challenges and Best Practices

1. **Planning and Testing:** Thorough planning and testing are crucial to ensure a smooth migration. This includes creating a detailed migration plan and conducting comprehensive testing in a staging environment.
2. **Configuration and Optimization:** Post-migration, it's essential to fine-tune ALB settings and monitor application performance to optimize the benefits of ALB features.
3. **Security Considerations:** Implementing best practices for security, such as HTTPS listeners and security groups, is vital to protect application data during and after migration.

Conclusion

Migrating from Classic ELB to Application Load Balancer presents a strategic opportunity for organizations to enhance their application architectures, improve performance, and leverage advanced features designed for modern applications. While the migration process requires careful planning and consideration, the benefits of ALB in terms of scalability, performance, and security make it a compelling choice for organizations looking to optimize their cloud infrastructure.

References

[1]. AWS Documentation on Migrating Your Classic Load Balancer. https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/migrate-classic-load-balancer.html
[2]. Migrate Classic Load Balancer to an ALB or NLB | AWS re:Post. https://repost.aws/knowledge-center/migrate-classic-load-balancer
[3]. Differences Between an ELB and an ALB on AWS - Sumo Logic. https://www.sumologic.com/blog/aws-elb-alb/
[4]. Upgrading Elastic Beanstalk Load Balancer from Classic to Application. https://serverfault.com/questions/960410/upgrading-elastic-beanstalk-load-balancer-from-classic-to-application-load-balan
[5]. ELB Vs ALB – How to Choose the Best AWS Load Balancer? – Cloudlytics. https://cloudlytics.com/elb-vs-alb-load-balancing-whats-the-difference/
[6]. AWS Elastic Load Balancer vs Application Load Balancer – FlareCompare. https://flarecompare.com/Cloud%20Networking/AWS%20Elastic%20Load%20Balancer%20vs%20Application%20Load%20Balancer/
[7]. AWS Migrate Classic Load Balancer to Application Load Balancer - Stack Overflow. https://stackoverflow.com/questions/47092254/aws-migrate-classic-load-balancer-to-application-load-balancer