IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15303

IMPLEMENTING INFRASTRUCTURE AS CODE WITH TARRAFORM AND DOCKER

A. Mohammed Almaas[1], Dr. C. Daniel Nesakumar[2]

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore – 641006, Tamil Nadu, India.[1]

Associate Professor, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore – 641006, Tamil Nadu, India.[2]

Abstract: The rapid growth of digital communication has increased the need for efficient methods of sharing web links across platforms. Long and complex URLs often reduce readability, are difficult to manage, and are not suitable for sharing in environments such as social media, emails, or printed documents. This paper presents **SkyLink**, a cloud-native URL shortening platform designed to generate concise and trackable links while providing analytics on user engagement.

SkyLink is developed using a modern full-stack architecture that combines **React and TypeScript** for the frontend and **Supabase** as a Backend-as-a-Service platform. The system enables users to generate shortened URLs, track click statistics, and manage links through an interactive dashboard. Containerization using Docker ensures portability, while Infrastructure as Code implemented through Terraform allows automated provisioning of cloud resources on AWS.

The system architecture emphasizes scalability, reliability, and ease of deployment. By integrating modern DevOps practices such as containerization, automated CI/CD pipelines, and serverless infrastructure, SkyLink demonstrates how contemporary web technologies can be used to build scalable and production-ready applications. Experimental evaluation shows that the system performs efficiently in generating short URLs and handling user requests with minimal latency. The proposed platform provides a cost-effective and flexible alternative to existing commercial URL shortening services.

I. INTRODUCTION

The internet has become the primary medium for communication, information exchange, and online services. As websites grow more complex, the URLs used to access them often become lengthy and difficult to share. Long URLs may contain multiple parameters, nested directories, and tracking information, making them visually unappealing and inconvenient for users.

URL shortening services were introduced to address this problem by converting long URLs into short, easy-to-share links. These shortened links not only simplify sharing but also enable tracking of user interactions such as click counts and engagement metrics. However, many existing commercial URL shortening platforms impose restrictions on analytics features or require subscription fees for advanced functionality.

To overcome these limitations, this project proposes **SkyLink**, a modern cloud-based URL shortening system designed to provide efficient link management and analytics without the constraints of proprietary platforms. The application leverages modern web technologies to ensure a responsive user interface, secure data storage, and scalable infrastructure deployment.

The system adopts a cloud-native architecture and integrates tools such as Docker for containerization and Terraform for infrastructure management. This approach enables automated deployment, scalability, and reliability while reducing operational complexity. The development of SkyLink demonstrates how modern software engineering practices can be applied to build efficient and scalable web services.

II. RELATED WORK

Several URL shortening platforms have been developed over the years to simplify link sharing and provide analytics features. Popular services such as **Bitly**, **TinyURL**, and **Rebrandly** provide tools for creating shortened links and monitoring their usage. These systems typically offer analytics features including click tracking, geographic data, and referral information.

DOI: 10.17148/IJARCCE.2026.15303

Bitly is one of the most widely used URL shortening platforms and provides comprehensive analytics and link management features. However, advanced capabilities such as branded domains and detailed analytics require paid subscriptions. TinyURL offers a simple interface for generating short links but lacks built-in analytics and advanced management tools. Rebrandly focuses on branded link management and marketing analytics, but many of its features are available only in premium plans.

Open-source alternatives such as **YOURLS (Your Own URL Shortener)** allow users to host their own URL shortening services. While these solutions provide greater control over data and infrastructure, they often rely on older technology stacks and require significant server configuration.

The limitations of existing systems highlight the need for a modern, scalable, and cost-effective URL shortening platform. The proposed SkyLink system addresses these limitations by combining modern frontend technologies, cloud-native deployment, and Infrastructure as Code to create a flexible and scalable solution.

III. OBJECTIVES AND CHALLENGES

Objectives

The primary objectives of the proposed SkyLink system are:

1. To design and develop a web-based platform capable of generating short URLs from long web links.
2. To provide real-time click tracking and analytics for shortened URLs.
3. To implement a scalable cloud-based infrastructure using containerization and Infrastructure as Code.
4. To create a user-friendly interface that simplifies link creation and management.
5. To ensure secure data storage and reliable system performance through modern backend technologies.

Challenges

Developing a cloud-based URL shortening service involves several technical challenges:

- **Scalability:** The system must handle a large number of requests and link redirections efficiently.
- **Unique Code Generation:** Ensuring that each shortened URL is unique without collisions.
- **Data Security:** Protecting user data and preventing unauthorized access.
- **Infrastructure Management:** Automating deployment and scaling using cloud technologies.
- **Performance Optimization:** Minimizing response time during URL redirection and analytics updates.

Addressing these challenges requires careful system design and the integration of reliable technologies.

IV. SYSTEM ARCHITECTURE

The SkyLink system follows a cloud-native architecture consisting of multiple layers that work together to provide URL shortening and analytics functionality.

The architecture includes the following components:

1. **Frontend Layer:**
   The user interface is developed using React and TypeScript, enabling a responsive and interactive web application.
2. **Backend Layer:**
   Supabase provides database services, authentication, and REST APIs for handling application data.
3. **Database Layer:**
   A PostgreSQL database stores information related to URLs, including original links, shortened codes, click counts, and creation timestamps.
4. **Containerization Layer:**
   Docker is used to package the application and its dependencies into containers, ensuring consistent execution across different environments.
5. **Cloud Infrastructure:**
   The application is deployed on AWS using services such as ECS Fargate, Application Load Balancer, and Elastic Container Registry.
6. **Infrastructure Management:**
   Terraform automates the creation and management of cloud resources, enabling reproducible deployments.

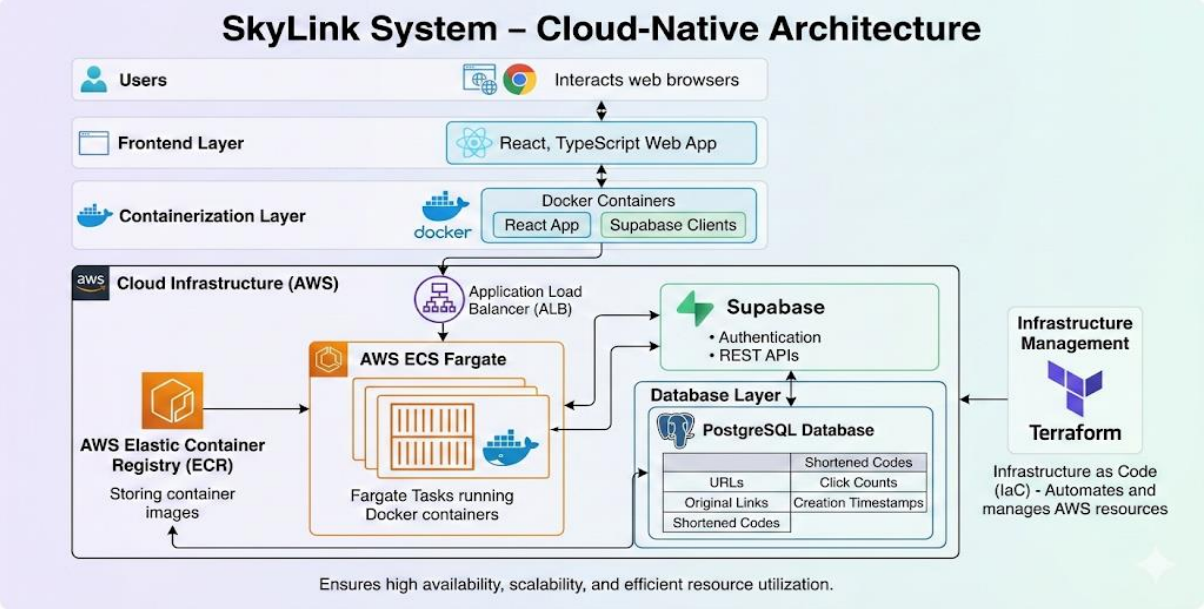
This architecture ensures high availability, scalability, and efficient resource utilization.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15303



V. IMPLEMENTATION

The SkyLink system is implemented using a modern full-stack development approach.
The frontend application is built using React and TypeScript, which provide a component-based architecture and improved code maintainability. The application interface includes a URL shortening form, a dashboard displaying analytics, and a list of recently created links.

The backend services are managed through Supabase, which provides a PostgreSQL database and RESTful APIs for storing and retrieving URL data. When a user submits a long URL, the system generates a unique short code and stores it in the database along with metadata such as creation time and click count.

The application is containerized using Docker to ensure consistent deployment across environments. Terraform scripts are used to provision cloud infrastructure automatically on AWS, including networking, container orchestration, and load balancing.

Additionally, a continuous integration and deployment pipeline is configured using GitHub Actions to automate testing, building, and deployment of the application.

VI. EVALUATION RESULTS AND DISCUSSIONS

The performance of the SkyLink system was evaluated based on its ability to generate shortened URLs, handle redirection requests, and display analytics data.

Testing results indicate that the system can generate short URLs within milliseconds and handle multiple user requests simultaneously. The click tracking mechanism successfully records each redirection event, enabling accurate analytics reporting.

The dashboard provides real-time statistics including the total number of links created, total clicks recorded, and average clicks per link. Visualization tools display the most frequently accessed links through graphical charts.

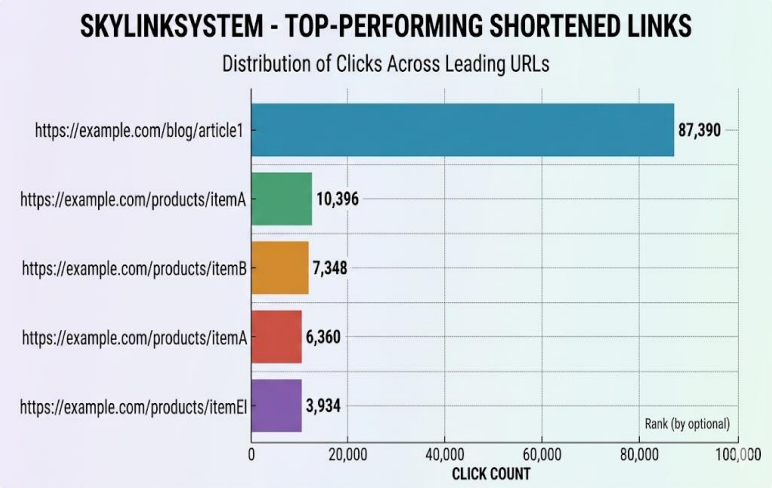
The use of containerized deployment and serverless infrastructure ensures efficient resource usage and improved system reliability. Overall, the evaluation demonstrates that the proposed system provides a scalable and efficient solution for URL shortening and analytics.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15303



VII. CONCLUSION

This paper presented SkyLink, a cloud-native URL shortening platform designed to simplify link sharing and provide analytics for user engagement. The system integrates modern web technologies and cloud infrastructure to create a scalable and efficient solution.

The implementation demonstrates how containerization, Infrastructure as Code, and cloud-based services can be combined to develop production-ready applications. The results show that the platform successfully generates shortened URLs, tracks click statistics, and provides an intuitive user interface for link management.

The proposed system offers a flexible and cost-effective alternative to traditional URL shortening services while maintaining high performance and scalability.

VIII. FUTURE ENHANCEMENTS

Although the current implementation of SkyLink provides essential URL shortening and analytics features, several improvements can be incorporated in the future:

1. Integration of user authentication and account management.
2. Support for custom branded domains for shortened links.
3. Advanced analytics including geographic and device-based statistics.
4. Implementation of rate limiting and security enhancements.
5. Development of a mobile application for easier access and link management.

These enhancements will further improve the functionality and usability of the system.

REFERENCES

[1]. Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures. University of California, Irvine.
[2]. Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*.
[3]. HashiCorp. (2023). Terraform: Infrastructure as Code.
[4]. PostgreSQL Global Development Group. PostgreSQL Documentation.
[5]. AWS Documentation. Amazon Web Services Cloud Architecture Guidelines.
[6]. Supabase Documentation. Open Source Firebase Alternative.

