

Infrastructure Automation and Continuous Deployment Solution

Andres Ricardo Martinez Diaz
David Alejandro Vasquez Carreño
Richard Santiago Urrea Garcia

December 2023

Contents

1	Introduction	3
1.1	Problem	3
1.2	Sample project infrastructure	3
1.3	Executive Summary	3
2	Concepts	4
2.1	AWS	4
2.2	SAM CLI	4
2.3	CDK	4
2.4	GitHub Actions	4
3	Overview	4
3.1	Prototype Architecture	4
4	Key Components	4
4.1	SAM for Deployment	4
4.2	Custom API for Template Guidance	5
4.3	AWS CDK for Template Generation	5
4.4	GitHub Actions for Continuous Deployment	5
4.5	Front-End Client for User Interaction	5
5	Application View	6
6	Business View	6
6.1	Executive Leadership	6
6.2	IT Operations Teams	6
6.3	Development Teams	7
6.4	Security Teams	7
6.5	Financial Stakeholders	7
6.6	Customers	7
7	Benefits	7
7.1	Streamlined Deployment Process	7
7.2	Collaborative Infrastructure Development	7
7.3	Efficient Continuous Deployment	8
7.4	User-Centric Front-End Interface	8
8	Use Cases	8
9	Conclusion	8

1 Introduction

1.1 Problem

We noticed a problem that happens very commonly in small companies or in companies that are making an adoption to the cloud and do not know how to manage their resources correctly. This problem consists of the lack of control and standards to manage infrastructure. This results in:

- Deploying resources manually and prone to errors.
- Complexity to effectively manage the infrastructure of cloud projects.
- Lack of standards in infrastructure management.
- Problems when configuring resources in the cloud.
- Lack of traceability of infrastructure changes within the companies.

1.2 Sample project infrastructure

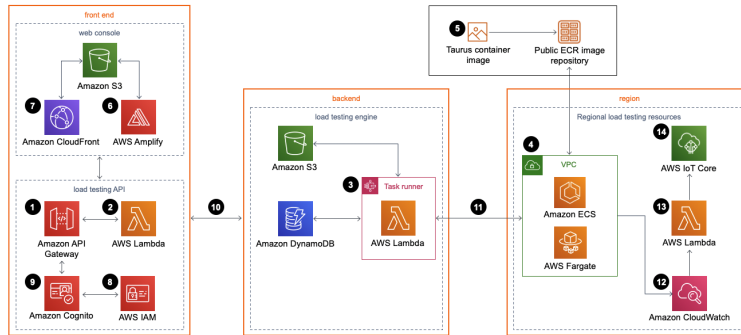


Figure 1: Sample Infrastructure

1.3 Executive Summary

In response to the growing complexity and dynamism of modern IT infrastructures, we have successfully implemented an Infrastructure Automation and Continuous Deployment Prototype Solution using Amazon Web Services (AWS) Cloud Development Kit (CDK), sam-cli, GitHub Actions, a customized API, and a user-friendly front-end client. This cutting-edge solution enables us to efficiently manage infrastructure changes, enhance development workflows, and ensure a seamless and reliable deployment process.

2 Concepts

2.1 AWS

AWS or Amazon Web Services is a cloud service provider, offering storage services, computing resources, applications, databases, etc.

2.2 SAM CLI

The AWS SAM CLI is a command line tool that you can use with AWS SAM templates and supported third-party integrations to build and run your serverless applications.

2.3 CDK

The AWS Cloud Development Kit (AWS CDK) is an open-source software development framework to define cloud infrastructure in code and provision it through AWS CloudFormation. It offers a high-level object-oriented abstraction to define AWS resources imperatively using the power of modern programming languages.

2.4 GitHub Actions

GitHub Actions is an automated workflow and continuous integration/continuous deployment (CI/CD) platform provided by GitHub. It allows defining and executing custom workflows directly in a GitHub repository. These workflows can automate various tasks, such as building, testing, and deploying code, in response to events like pushes, pull requests, or other GitHub activities.

3 Overview

The primary goal of this solution is to streamline the management of AWS infrastructure by leveraging powerful automation tools and cloud-native services. By integrating sam-cli into our deployment pipeline, we have achieved a high degree of automation in provisioning, updating, and scaling cloud resources. Additionally, GitHub Actions serve as the orchestrator for this automation, enabling a seamless integration of infrastructure changes with our codebase.

3.1 Prototype Architecture

4 Key Components

4.1 SAM for Deployment

AWS SAM is at the core of our deployment strategy, facilitating the deployment of AWS resources.

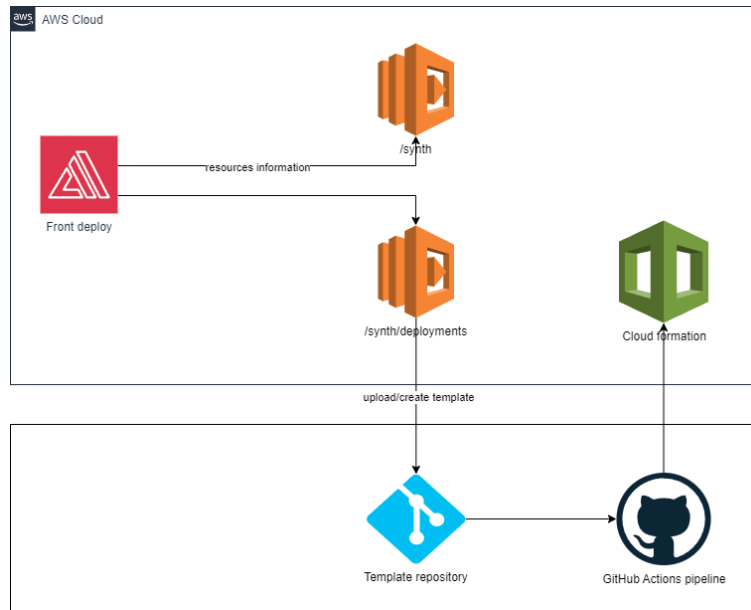


Figure 2: Prototype Architecture

4.2 Custom API for Template Guidance

The custom API acts as a knowledge hub for creating templates, providing the tools for structuring and configuring stacks.

4.3 AWS CDK for Template Generation

The AWS CDK takes the lead in generating infrastructure templates. With the help of the API, it uses the received data to define or update a stack using a programming language.

4.4 GitHub Actions for Continuous Deployment

GitHub Actions has been seamlessly integrated into our workflow, automating the continuous deployment process. This includes validating changes, running tests, and deploying updates to the stacks using SAM. The integration with GitHub Actions ensures a robust and efficient CI/CD pipeline.

4.5 Front-End Client for User Interaction

The front-end client serves as an intuitive interface, simplifying the interaction with new or updated stacks. Users can effortlessly manage and monitor the deployment process, providing both technical and non-technical stakeholders with a user-friendly experience.

5 Application View

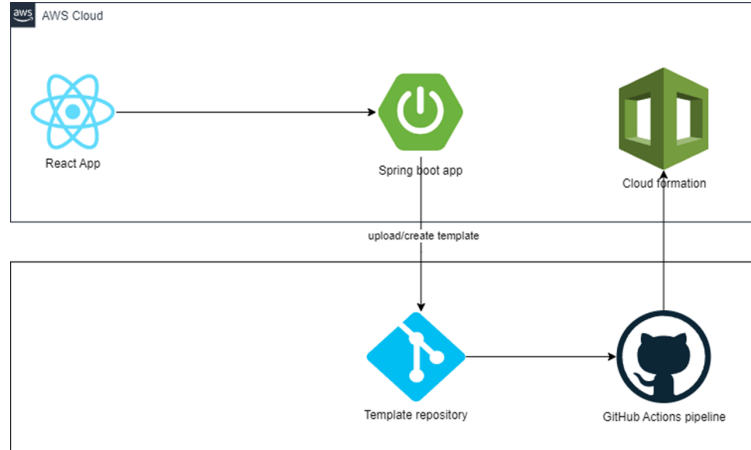


Figure 3: Apps Architecture

6 Business View

The project's business view emphasizes a strategic shift towards modern, automated infrastructure deployment. This positions the organization for future growth by fostering agility, reliability, and cost-effectiveness in its technology operations. The value proposition for stakeholders highlights tangible benefits across various facets of the business, making it a strategic investment for sustained success.

6.1 Executive Leadership

Value Proposition: Accelerated Time-to-Market

By automating infrastructure deployment, the project significantly reduces the time required to bring new features or products to market. This aligns with business goals of staying ahead in a fast-paced industry.

6.2 IT Operations Teams

Value Proposition: Improved Operational Efficiency

The automated deployment process reduces the burden on IT operations, minimizing manual interventions, and mitigating the risk of errors. This results in smoother operations, reduced downtime, and improved resource utilization.

6.3 Development Teams

Value Proposition: Enhanced Developer Productivity

Developers benefit from faster feedback cycles, enabling them to iterate quickly. Automation frees up developers from manual deployment tasks, allowing them to focus on writing code and delivering business value.

6.4 Security Teams

Value Proposition: Consistent Security Measures

The use of SAM, AWS CDK, and GitHub Actions ensures that security measures are consistently applied across environments. Security teams can rely on standardized deployment processes, reducing the risk of misconfigurations.

6.5 Financial Stakeholders

Value Proposition: Cost Optimization

Automation not only accelerates deployment but also contributes to cost optimization. Efficient resource scaling and the ability to handle varying workloads reduce unnecessary resource consumption, aligning with financial goals.

6.6 Customers

Value Proposition: Enhanced Service Reliability

Faster deployments and reduced downtime contribute to a more reliable service for customers. Continuous improvements, facilitated by automated deployments, lead to a better overall customer experience.

7 Benefits

7.1 Streamlined Deployment Process

The combined power of SAM, CDK, and GitHub Actions streamlines the deployment process, reducing manual intervention and speeding up the delivery of changes to the infrastructure.

7.2 Collaborative Infrastructure Development

Infrastructure as code with CDK fosters collaboration and version control, while the custom API provides real-time guidance. This collaborative approach ensures that infrastructure changes align with best practices and project requirements.

7.3 Efficient Continuous Deployment

GitHub Actions automates the continuous deployment pipeline, providing a reliable mechanism for validating, testing, and deploying changes. This results in increased efficiency and a faster feedback loop for developers.

7.4 User-Centric Front-End Interface

The front-end client enhances user experience, making it convenient for both technical and non-technical users to manage and monitor the deployment of new stacks or updates. This focus on user-centric design promotes usability and accessibility.

8 Use Cases

1. Serverless Application Deployment.
2. Infrastructure as Code (IaC) Updates
3. Multi-Region Deployments
4. Custom Resource Management
5. Infrastructure Updates Based on Pull Requests
6. Artifact Management
7. Continuous Learning and Documentation

9 Conclusion

Our integrated solution represents a leap forward in infrastructure management, combining the strengths of SAM, CDK, GitHub Actions, and a user-friendly front-end. By leveraging these tools synergistically, we have created an ecosystem that not only accelerates deployment but also enhances collaboration, reliability, and the overall user experience in managing AWS stacks.

References

- [1] Amazon Web Services. (n.d.). AWS. Retrieved from <https://aws.amazon.com>
- [2] AWS. (n.d.). AWS Serverless Application Model Command Line Interface (AWS SAM CLI). Retrieved from <https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/what-is-sam.html>
- [3] AWS. (n.d.). AWS Cloud Development Kit (AWS CDK). Retrieved from <https://docs.aws.amazon.com/cdk/v2/guide/home.html>

- [4] GitHub. (n.d.). GitHub Actions. Retrieved from <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>
- [5] SAM for Deployment. (n.d.). Retrieved from <https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/what-is-sam.html>
- [6] AWS example architecture. (n.d.). Retrieved from <https://docs.aws.amazon.com/solutions/latest/distributed-load-testing-on-aws/architecture.html>