



International Journal of Multidisciplinary Research and Growth Evaluation



International Journal of Multidisciplinary Research and Growth Evaluation

ISSN: 2582-7138

Received: 21-05-2021; Accepted: 16-06-2021

www.allmultidisciplinaryjournal.com

Volume 2; Issue 3; May-June 2021; Page No. 562-566

Exploring AWS Deployment Strategies

Siva Kumar Mamillapalli

Independent Researcher, USA

Corresponding Author: Siva Kumar Mamillapalli

DOI: <https://doi.org/10.54660/IJMRGE.2021.2.3.562-566>

Abstract

Amazon Web Services (AWS) offers a comprehensive suite of deployment options that can support the full lifecycle of your application infrastructure. Whether your needs are as simple as a straightforward three-tier web application or as complex as a large-scale, multi-service architecture with intricate workloads, AWS provides the tools, resources, and services to manage and deploy your applications efficiently. AWS's deployment options are designed to scale and adapt to the growing and dynamic needs of businesses of all sizes, making it possible to tailor deployment strategies to both technical and organizational requirements.

This paper provides a detailed examination of the core features available in these services, such as automation,

scalability, and flexibility. In addition, it delves into the most effective strategies for deploying application stacks—whether you're deploying from scratch, managing existing applications, or continuously updating systems.

Furthermore, the paper explores best practices for application deployment, touching on aspects such as infrastructure-as-code (IaC), containerization, continuous integration/continuous deployment (CI/CD), and managing deployment updates seamlessly with minimal downtime. By understanding these deployment strategies and leveraging the right AWS tools, organizations can optimize their application delivery, ensure smoother updates, and enhance the reliability of their infrastructure.

Keywords: AWS, Cloud Deployment, CloudFormation, IaC, CI/CD, DevOps, AWS CDK

1. Introduction

Designing a scalable, efficient, and cost-effective deployment solution goes beyond updating application versions. It involves managing the entire application lifecycle, including resource provisioning, configuration management, software updates, monitoring, access control, and more. AWS offers various services that cater to these needs, allowing organizations to choose between manual resource management or leveraging AWS-managed services for convenience. Depending on the application's requirements, these services can be used individually or combined to create robust deployment solutions, accelerating application development and delivery with reliability.

2. Literature Review

2.1 AWS CloudFormation

AWS CloudFormation is a service that allows customers to provision and manage nearly any AWS resource using custom templates in YAML or JSON. These templates define infrastructure resources within a group called a stack, enabling users to customize and control all components necessary for their application while maintaining full oversight. By using templates, you gain the ability to implement version control on your infrastructure and quickly replicate it reliably. AWS CloudFormation provides detailed control over both low-level components like route tables and subnets, as well as higher-level components like CloudFront distributions. It is often used in conjunction with other AWS deployment services or third-party tools to manage application code deployments on infrastructure.

Table 1: AWS CloudFormation deployment features

Capability	Description
Provision	CloudFormation automatically provisions and updates the infrastructure components specified in a template.
Configure	AWS CloudFormation templates provide great flexibility for customizing and updating all infrastructure components.
Deploy	Modify your AWS CloudFormation templates to change the resources within a stack. Depending on your application's architecture, you may require an additional deployment service to update the application version running on your infrastructure.
Scale	AWS CloudFormation will not automatically ally handle infrastructure scaling on your behalf; however, you can configure auto scaling policies for your resources in a AWS CloudFormation template.
Monitor	AWS CloudFormation provides native monitoring of the success or failure of updates to infrastructure defined in a template, as well as drift detection to monitor when resources defined in a template do not meet specifications. Additional monitoring solutions will need to be in place for application-level monitoring and metrics.

The following diagram illustrates a typical use case for AWS CloudFormation. In this scenario, AWS CloudFormation templates are used to define all the necessary infrastructure components for a simple three-tier web application. The example shows how bootstrap scripts within AWS CloudFormation are employed to deploy the latest version of the application onto Amazon EC2 instances. However, it is also common to combine other deployment services with

AWS CloudFormation, using it solely for infrastructure management and provisioning. Multiple AWS CloudFormation templates are utilized to create the infrastructure, including IAM roles, VPCs, subnets, route tables, security groups, and S3 bucket policies. Separate templates are employed to build each part of the application architecture.

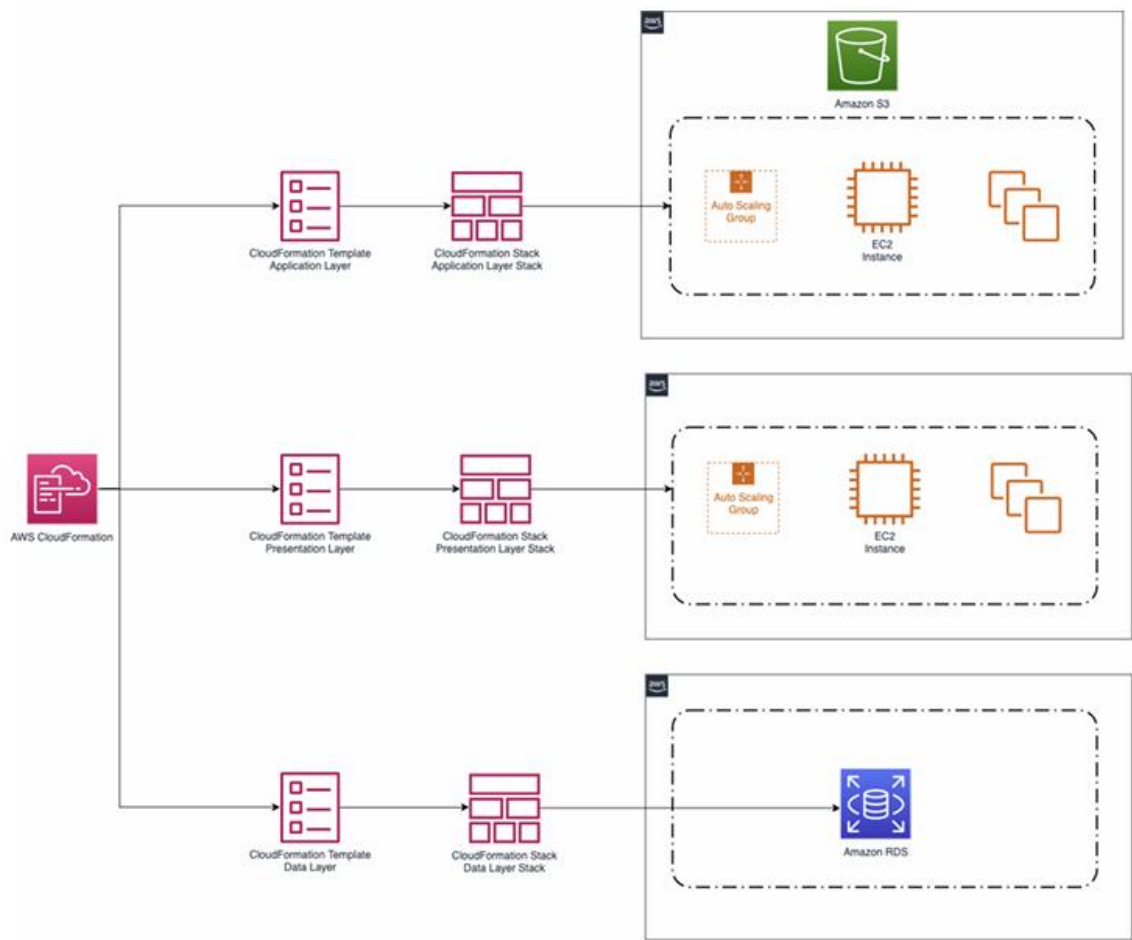


Fig 1: CloudFormation use case

2.2 AWS elastic beanstalk

AWS Elastic Beanstalk simplifies the deployment and scaling of web applications and services developed in various languages like Java, NET, Python, and more, on familiar servers such as Apache, Nginx, and others. It provides a comprehensive application management solution by handling

all infrastructure and platform tasks automatically. With Elastic Beanstalk, you can swiftly deploy, manage, and scale applications without the operational overhead of infrastructure management. This makes it ideal for organizations new to AWS or looking to deploy web applications quickly.

When using Elastic Beanstalk, you simply upload your source code and it takes care of provisioning servers, databases, load balancers, networks, and auto scaling groups. While it manages these resources, you maintain full control for customization. Elastic Beanstalk also ensures compliance with ISO, PCI, SOC 1, SOC 2, SOC 3, and HIPAA standards, enabling applications to handle regulated financial data or protected health information (PHI).

Table 2: AWS Elastic Beanstalk deployment features

Capability	Description
Provision	Elastic Beanstalk will create all infrastructure components necessary to operate a web application or service that runs on one of its supported platforms. If you need additional infrastructure, this will have to be created outside of Elastic Beanstalk.
Configure	Elastic Beanstalk provides a wide range of options for customizing the resources in your environment.
Deploy	Elastic Beanstalk automatically handles application deployments and creates an environment that runs a new version of your application without impacting existing users.
Scale	Elastic Beanstalk uses Elastic Load Balancing and Auto Scaling to automatically scale your application in and out based on its specific needs. Multiple availability zones give you an option to improve application reliability and availability.
Monitor	Elastic Beanstalk offers built-in environment monitoring for applications including deployment success/failures, environment health, resource performance, and application logs.
Graviton support	AWS Graviton arm64-based processors deliver the best price performance for your cloud workloads running in Amazon EC2. With AWS Graviton on Elastic Beanstalk, you can select Amazon EC2 instance types to meet optimization needs of your workloads and benefit from improved price performance over a comparable x86-based processor.

Elastic Beanstalk simplifies the deployment and management of web applications on AWS. For example, it can be used to quickly deploy a basic web application, with all necessary infrastructure—such as security groups, IAM roles, and CloudWatch alarms—handled by Elastic Beanstalk. It automatically provisions Amazon EC2 instances with the required runtime environments and deployment packages. Additionally, Elastic Beanstalk environments can integrate with resources like Amazon Relational Database Service (Amazon RDS) that are created independently of Elastic Beanstalk.

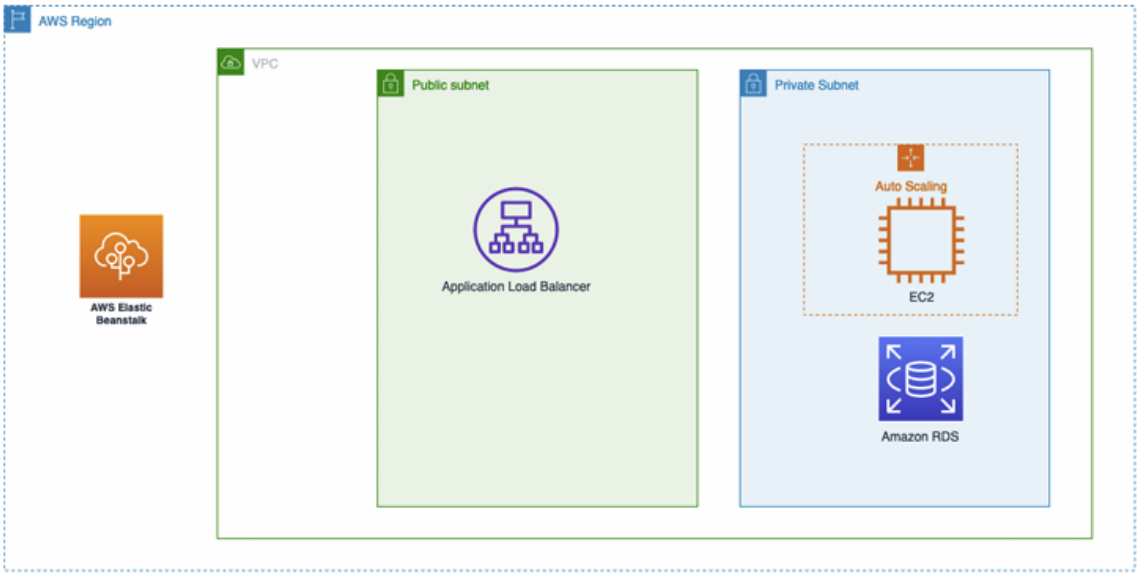


Fig 2: AWS Elastic Beanstalk use case

2.3 AWS CodeDeploy

AWS CodeDeploy is a fully managed service that automates application deployments to compute services like Amazon EC2, Amazon Elastic Container Service (Amazon ECS), AWS Lambda, or on-premises servers. It allows organizations to automate application deployments and eliminate error-prone manual processes. CodeDeploy supports a variety of application content, including code, serverless functions, configuration files, and more.

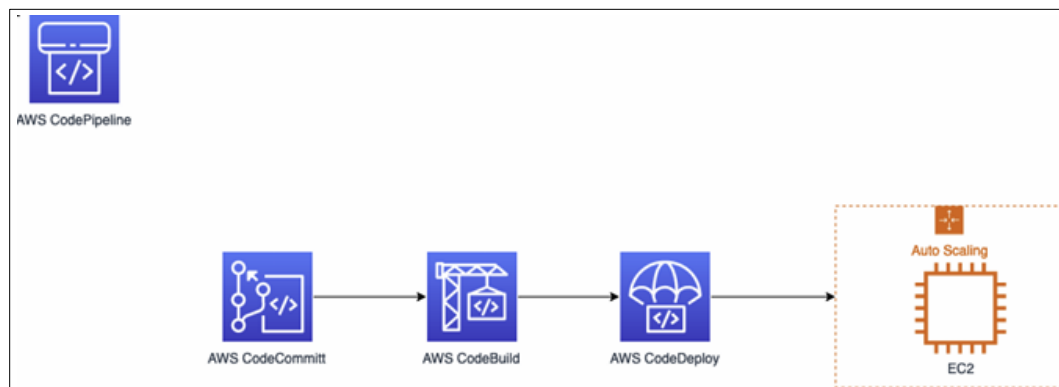
Designed as a building block service, CodeDeploy helps application developers deploy and update software on existing infrastructure. It is not an end-to-end application management solution and should be used alongside other AWS deployment services like AWS CodeStar, AWS CodePipeline, and third-party services to form a complete CI/CD pipeline. Additionally, CodeDeploy does not manage resource creation for users.

Table 3: AWS CodeDeploy deployment features

Capability	Description
Provision	CodeDeploy is designed to work with your existing compute resources and does not create resources on your behalf. It requires these resources to be organized into a structure known as a deployment group to deploy application content.
Configure	CodeDeploy uses an application specification file to specify customizations for compute resources.
Deploy	CodeDeploy provides different deployment strategies based on the type of compute resource it is used with.
Scale	CodeDeploy does not handle the scaling of your underlying application infrastructure; however, depending on your deployment configurations, it may create additional resources to facilitate blue/green deployments.
Monitor	CodeDeploy can track the success or failure of deployments and provides a history of all deployments, but it does not offer performance or application-level metrics.

The following diagram demonstrates a typical use case for CodeDeploy within a complete CI/CD solution. In this example, CodeDeploy works alongside other AWS Developer Tools—such as AWS CodePipeline (to automate CI/CD pipelines), AWS CodePipeline (to automate CI/CD pipelines), AWS CodeBuild (to build and test application components), and AWS CodeCommit (as the

source code repository)—to deploy an application to a group of Amazon EC2 instances. CodeDeploy handles the deployment of application components to compute resources within a deployment group, while all infrastructure components are set up outside of CodeDeploy.

**Fig 3:** AWS CodeDeploy use case

3. Conclusion

Deploying applications on AWS isn't a one-size-fits-all approach. When designing a deployment solution, it's crucial to tailor it to the type of application which determines the most suitable AWS services. Often, achieving comprehensive functionality for provisioning, configuring, deploying, scaling, and monitoring applications requires integrating multiple deployment services. A typical strategy involves using CloudFormation and its extensions for managing general infrastructure, while employing specialized solutions for application updates. For containerized applications, CloudFormation sets up the infrastructure, while Amazon ECS or Amazon EKS handles container provisioning, deployment, and monitoring. Furthermore, AWS deployment services can be integrated with third-party tools, facilitating seamless incorporation into existing CI/CD pipelines or infrastructure management solutions. For instance, OpsWorks synchronizes configurations between on-premises and AWS nodes, and CodeDeploy collaborates with various third-party CI/CD services to complete deployment pipelines. AWS offers a range of tools to streamline and automate infrastructure provisioning and application deployment, with each deployment service providing distinct capabilities for managing applications. To create an effective deployment architecture, assess the features of each service in relation to the specific requirements of your application and organization.

4. References

1. Amazon Web Services. How Energy and Utility Companies Can Recover from Ransomware and Other Disasters Using IaC on AWS. Amazon Web Services; 2021 Jan 31. Available from: <https://aws.amazon.com/blogs/industries/how-energy-and-utility-companies-can-recover-from-ransomware-and-other-disasters-using-iac-on-aws/>
2. Artac M, Borovšak T, Di Nitto E, Guerriero M, Perez-Palacin D, Tamburri DA. Infrastructure-as-Code for Data-Intensive Architectures: A Model-Driven Development Approach. In: 2018 IEEE International Conference on Software Architecture (ICSA). IEEE; 2018 Apr. p. 156-1609.
3. Bhandari P. Security Best Practices in Infrastructure as Code. In: Proceedings of the International Conference on Cybersecurity and Privacy; 2020. p. 88-95.
4. Bhandari P. Infrastructure as Code Security and Best Practices | A Quick Guide. Xenonstack; 2020 May 4. Available from: <https://www.xenonstack.com/insights/infrastructure-as-code-security>
5. Davis R. Static Analysis for Infrastructure as Code. Security & Privacy Journal. 2019;22(4):45-52.
6. Farias J, Albornoz S. Netflix Cloud Architecture: A Case Study in Scalability and Flexibility. Journal of Cloud Computing. 2020;9(1):15-27.
7. HashiCorp. HashiCorp Infrastructure Survey 2019. HashiCorp; 2019.

8. Trend Micro. Infrastructure as Code: Security Risks and How to Avoid Them. Trend Micro; 2020. Available from: <https://www.trendmicro.com/vinfo/in/security/news/virtualization-and-cloud/infrastructure-as-code-security-risks-and-how-to-avoid-them>
9. Jones L. Implementing Pre-Commit Hooks for Secure Codebases. *Software Engineering Practices*. 2020;15(3):102-110.
10. Amazon Web Services. Overview of Deployment Options: AWS CloudFormation. Available from: <https://docs.aws.amazon.com/whitepapers/latest/overview-deployment-options/aws-cloudformation.html>. Accessed Feb 2021.
11. Amazon Web Services. Overview of Deployment Options: AWS Elastic Beanstalk. Available from: <https://docs.aws.amazon.com/whitepapers/latest/overview-deployment-options/aws-elastic-beanstalk.html>. Accessed Feb 2021.
12. Amazon Web Services. Overview of Deployment Options: AWS CodeDeploy. Available from: <https://docs.aws.amazon.com/whitepapers/latest/overview-deployment-options/aws-codedeploy.html>. Accessed Mar 2021.