International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 02 | Feb - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Seamless Transition and Governance of Unmanaged AWS Resources Using IAC Code Generator Service Automation

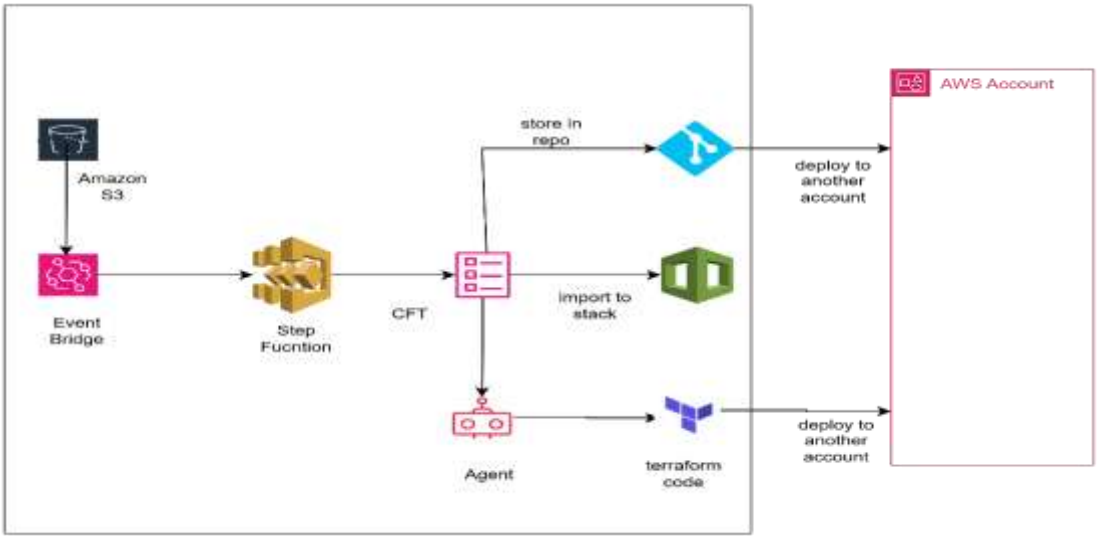
Monalisa kaur, Shubhankar Datta

Introduction

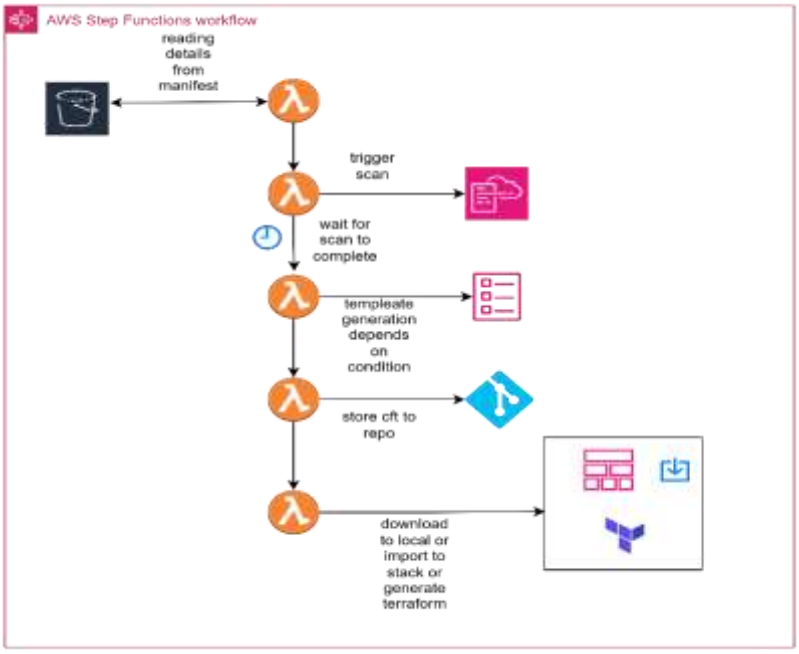
In modern cloud environments, efficiently managing infrastructure and ensuring consistency across multiple accounts and regions are critical challenges, especially when dealing with manual processes or unmanaged resources. The approach described here leverages automation to streamline Infrastructure as Code (IaC) generation and management, transforming how cloud resources are provisioned, tracked, and maintained in AWS environments. This solution enhances governance, compliance, and developer productivity while eliminating the inconsistencies and manual effort associated with traditional resource management. It empowers organizations to manage infrastructure at scale with automation, ensuring faster, more reliable infrastructure management and reducing the risks tied to manual configuration changes.

This facilitates bulk updates, compliance enforcement, and consistent configuration management across all resources. The generated templates can be imported as AWS CloudFormation stacks, ensuring that all resources adhere to the same standards and policies, allowing for easy configuration changes across a large number of resources.

Architecture Diagram:



Overall architecture diagram



Step Function Architecture Diagram

Prerequisite :

IAM permissions required for scanning resources

To scan resources with IaC generator, your IAM principal (user, role, or group) must have:

- CloudFormation scanning permissions
- Read permissions for target AWS services

The scan scope is limited to resources you have read access to. Missing permissions won't cause scan failure but will exclude those resources.

Process Overview:

1. Deploy a base CloudFormation stack that creates:

- An S3 bucket (for manifest.json uploads)
- An EventBridge rule (listening to ObjectCreated events from the bucket)
- A Step Functions state machine
- Required IAM roles and Lambda functions

2.User prepares a manifest.json file containing:

- Target AWS account ID
- Target region for scan
- Optional resource scope
- Template generation filters
- Output storage location
- Final action (download / import stack / generate Terraform)

- Sample Manifest File :

manifest.json

3.User uploads manifest.json to the dedicated S3 bucket created during deployment.

4.S3 emits an ObjectCreated event.

5.EventBridge rule captures the event and triggers the Step Functions state machine.

6.Step Function execution starts with the S3 object details as input.

- Lambda 1 – Read Manifest
  - Reads manifest.json from S3
  - Validates schema
  - Passes parsed manifest data to next state
- Lambda 2 – Trigger Scan
  - Assumes role into target account (if cross-account)
  - Initiates IaC scan in specified region
  - Passes taskToken to scanning system
  - Step Function uses .waitForTaskToken
  - Workflow pauses
  - Scan runs asynchronously.
  - When scan completes:
  - Scan service (or callback Lambda) calls:
  - SendTaskSuccess(taskToken)
  - Step Function resumes automatically
- Lambda 3 – Template Generation
  - Reads scan results
  - Applies filter logic:
  - Tag-based
  - Resource-type-based
  - Identifier-based
  - Generates CloudFormation template structure
- Lambda 4 – Store Template
  - Stores generated CFT in repository/location defined in manifest
  - Could be:
  - S3 path
  - CodeCommit repo
  - GitHub repo
- Lambda 5 – Final Action Handler
  - If download → generate pre-signed URL
  - If import_stack → call CloudFormation Create/Import Stack
  - If terraform → invoke Bedrock agent to generate Terraform code
  - Return final output status
- Step Function execution completes successfully.
- Any failure in any step of the workflow will result in the immediate failure of the Step Functions execution

USE Cases:

- **Use case 1: Sandbox-to-Dev/QA Automated Infrastructure Promotion**
  - **Use Case Overview:**

In shared sandbox environments, developers freely create and validate cloud resources, but there is no structured mechanism to reliably identify, capture, and replicate approved configurations into development environments. This results in manual rework, inconsistencies, and lack of traceability when promoting infrastructure across accounts or regions.

  - **How Our Solution Fits:**

Our event-driven IaC automation solution periodically discovers and filters sandbox resources based on defined policies, automatically generates infrastructure templates, and stores them in a controlled repository. These approved templates can then be deployed to dev or other target environments, enabling consistent, repeatable infrastructure promotion while significantly reducing manual effort and supporting bulk updates at scale.

- **Use case 2: Automated IaC Onboarding for Existing AWS Resources**
  - **Use Case Overview:**

Many AWS resources exist outside of IaC management, making bulk updates, compliance enforcement, and consistent configuration challenging. Manually identifying and standardizing these unmanaged resources is time-consuming and error-prone, leading to governance gaps across accounts and regions.

  - **How Our Solution Fits:**

Our solution automatically discovers unmanaged resources, filters them based on policy criteria, and generates IaC templates capturing their current configuration. These templates are stored in version-controlled repositories and can be imported into managed stacks, enabling bulk updates, enforcing standards, and supporting consistent infrastructure management across all resources.

Conclusion:

This solution establishes a fully automated, event-driven workflow that orchestrates Infrastructure as Code (IaC) scanning, template generation, and controlled deployment using AWS Step Functions and Lambda. Through a structured manifest-driven approach and asynchronous execution model, it enables scalable and consistent infrastructure management across multiple accounts and regions.

By eliminating manual processes and standardizing template generation, the solution enhances governance, improves operational efficiency, and supports bulk updates across resources, ensuring streamlined and reliable infrastructure management at scale.