FPAC CI/CD Automation Overview

This document defines the secure, least-privilege architecture used by the FPAC CI/CD pipeline for deploying AWS infrastructure using the AWS Cloud Development Kit (CDK). It provides a detailed explanation of the identity model, idempotency requirements, and the automated deployment workflow that ensures consistent, compliant, and secure cloud operations.

Executive Summary

The FPAC CI/CD pipeline deploys all cloud application components using a tightly scoped, least-privilege security model built on the AWS Cloud Development Kit (CDK). This model ensures that:

- Infrastructure is defined and provisioned exclusively through IaC (no console changes).
- All deployments are idempotent, deterministic, and safe to re-run.
- Permissions for CI/CD are minimized, reducing scope.
- All modifications are version-controlled and automated.
- CloudFormation acts as the only provisioning engine, with restricted permissions.

The result is a secure, automated, repeatable deployment framework aligned with FPAC and USDA security requirements.

Background - current state

Current deployments into the FPAC AWS environment rely heavily on manual steps performed by individual users. This approach introduces operational and security challenges, including:

- Inconsistent configuration across environments (DEV/TEST/CERT/PROD)
- Limited repeatability and auditability of changes
- Increased risk of human error during deployments
- Delays in implementing security fixes and patches
- Requires individuals to have AWS Console access, because there is no alternative for implementing solutions
- There is currently NO guarantee that a solution tested in a lower environment, is the solution that is deployed in a higher environment
- Difficult to spin-up new environments
- Difficult to roll-back a release

- Difficult to provide consistent hot-fix testing environment based on current state of production
- Binary tooling is utilized with no change control or deployment process
- No ability to run industry standard scanning tools on source code prior to deployment

1. Architecture Overview

The architecture uses a two-policy model:

1.2 Deployer Policy attached to an IAM User

Used by the CI/CD pipeline to:

- Synthesize CDK templates
- Create/Update/Delete CloudFormation stacks
- Pass a restricted execution role to CloudFormation
- Perform read-only environment lookups

Critical:
This IAM user does NOT have permission to create AWS resources directly.

1.3 Execution Policy (CDK-EXECUTOR)

Assumed by CloudFormation during deployment.
This is the only policy permitted to:

- Create application infrastructure
- Modify AWS services approved by FPAC
- Delete resources during stack updates
- Work within a restricted service and resource set

This policy enforces the allowed services list and prevents creation of unauthorized resources (e.g., EC2 unless explicitly enabled).

1.4 Idempotency and CDK Requirements

The architecture assumes:

- All infrastructure is defined in CDK.
- CDK code is idempotent—safe to run repeatedly without manual resets.
- No resource is created or modified outside the pipeline.
- CDK synthesizes CloudFormation templates that fully describe desired state.

Manual console changes are prohibited because they break idempotency and lead to drift.

All infrastructure must be deployed exclusively through CDK pipelines.

1.5 Solution Summary

The FPAC team is transitioning to an automated CI/CD model using Bitbucket and AWS-native services. To comply with least-privilege principles, the CI/CD pipeline requires an IAM User with a very limited scope AWS policy. This IAM user will be known as the deployer. The specific policy for this user is defined in the section 7.1, "Deployer IAM User AWS Policy".

The solution also requires a purpose-built AWS CDK execution policy with permissions needed to deploy CloudFormation stacks and manage application resources. The Cloudformation outputs will be generated using the AWS CDK, which is the tooling that will be utilized for the CI/CD pipelines. This second policy is defined in the section 7.2, "AWS CDK Execution Policy". This policy is applied during the AWS CDK bootstrap process, which is required for each account and region. The specific commands to perform this bootstrap process are defined in the section 7.3, "AWS CDK Bootstrapping"

2. Current Limitations

Under existing restrictions, the CI/CD pipeline does not have sufficient permissions to:

- Create or update CloudFormation stacks that encapsulate approved infrastructure-as-code patterns
- Pass pre-approved execution roles to CloudFormation
- Interact with S3 and KMS for encrypted pipeline artifacts
- Update specific Lambda and Glue resources associated with an application release
- Enable, configure and update various AWS services that are part of the Data pipeline process

As a result, deployments either cannot be fully automated, or they require manual intervention by privileged users, which increases operational risk and reduces traceability. It also requires many more

priviledged AWS console users than is generally required for solutions that have implemented robust and automated CI/CD pipelines.

3. Security Considerations and Alignment

This request is designed to comply with federal security requirements and FPAC governance standards, including:

- NIST SP 800-53 AC-6 (Least Privilege) – Access is limited to the minimum set of actions required for the CI/CD pipeline to perform its function.
- Separation of Duties – OCIO/DISC retains sole control over creation and modification of IAM console access roles, network infrastructure, VPCs, and EC2 services.
- Auditability – All changes occur via CloudFormation and CI/CD tooling, producing a consistent audit trail of deployments.

3.1 Scope and Limitations

- The solution is only acessed via the AWS CDK and is implemented using the AWS CDK
- The AWS CDK is an IaC framework, the deployment code will be in BitBucket repositories
- It does not grant console access to human users.
- It does not permit creation or modification of IAM users, groups, or customer-managed policies.
- It does not permit creation or alteration of VPCs, subnets, gateways, EC2 instances, or similar compute/network resources.
- It does not modify AWS Organizations

Any future expansion of scope would require a separate review and approval.

4. Benefits

Approving this solution will:

- Reduce deployment risk by minimizing manual, ad-hoc changes in the console
- Guarantee consistency between environments via repeatable, automated deployments
- Ensure that only tested solutions are deployed
- Provide the capability to roll-back a release
- Capabilities for testing hot-fixes to production by having staging environment

- Increase auditability by centralizing changes through CloudFormation and CI/CD logs
- Maintain strong security boundaries by enforcing least privilege
- Ultimately reduce the number of users that have ANY AWS console access and significantly reduce the privileges granted to those who retain access

This approach strengthens, rather than weakens, FPAC's overall security posture.

4.1 Final Summary Statement

This architecture defines a strict, least-privilege deployment model for FPAC cloud applications using the AWS Cloud Development Kit (CDK). By enforcing fully idempotent and automated deployments, the FPAC CI/CD pipeline ensures that all infrastructure changes are deterministic, reproducible, and version-controlled. No manual modifications to the AWS environment are permitted, eliminating drift and strengthening security.

The two-policy model—consisting of a limited Deployer and a tightly scoped Execution policy which minimizes the scope of CI/CD credentials while enabling CloudFormation to deploy only approved AWS services. This approach provides secure, auditable, and consistent provisioning of cloud infrastructure and aligns with FPAC's operational, compliance, and modernization objectives.

5. Transitioning to the Automated Process

There will need to ba a transition from the current software solution implementation, to the automated CI/CD pipeline approach. The transition assumes that the current operations, can't be interrupted. Thus, the new automated deployments will need to run along-side the existing solution.

5.1 Current Infrastructure

The current solution has two AWS accounts with services running in the us-east-1 region. The current implementation has DEV and CERT environments sharing an AWS account, with PROD on it's own AWS account.

5.2 New Accounts

The optimal solution will have two new AWS accounts, one for DEV and one for STAGING. These new accounts will always utilize the automated CI/CD process. It is assumed that these new accounts would have services deployed on us-east-1 region.

5.3 Transition

There are over thirty data pipelines, each has it's own BitBucket repository. The transition plan will be to rebuild each of these data pipelines as CDK applications. These CDK applications will be put in a BitBucket mono-repo, which will allow common CDK constructs to be shared across the data pipeline applications. These CDK applications will be deployed to the new DEV account and tested on the new STAGING account. The production deployment will be to us-east-2 region on the existing PROD account.

5.4 CI/CD Deployment Pipelines

The mono-repo in BitBucket, currently called FPAC-CDK, will have three CDK data application CI/CD pipelines. There will be a pipeline to deploy to the DEV environment, a pipeline to deploy to STAGING and a pipeline to deploy to PROD. These pipelines will utilize the IAM deployer user to run the cdk cli and deploy resources.

The mono-repo will also have three infrastructure pipelines, for DEV, STAGING and PROD. These will be used to deploy shared resources, such as S3 buckets, glue databases and other shared AWS services.

The deployment pipelines will be implemented in Jenkins, using webhooks from Jenkins, which are called by BitBucket based on specific branch, merge or tagging operations. Since there are dev, cert and prod Jenkins environments, the pipelines will utilize these environments via different web hooks. It is assumed that SSH and SSH keys can be shared from Jenkins to BitBucket, allowing Jenkins to pull the branch or tag from the mono-repo via SSH. The details of the git process, configuration and release management will be documented in a separate document.

5.5 Post Transition

Once all of the data pipeline applications have been transitioned to the CI/CD approach and architecture, the existing DEV resources should be removed from current DEV/CERT us-east-1 AWS account. The DEV/CERT account will transition to a dedicated CERT environment, which will be re-built, using the CI/CD pipeline and architecture. New pipelines will be available for the dedicated CERT environment. The PROD environment will have a mix of solutions on us-east-2 and us-east-1 and a dedicated transition plan will need to be formulated. The STAGING environment will become the pre-production deployment environment, which can be used to both preview a release as well as support testing for hot-fixes after a release has been deployed.

5.6 Additional Infrastructure

The clean split of resources utilizing new AWS accounts, might encounter unanticipated challenges. Thus, the us-east-2 region on the current DEV/CERT account will also be utilized to ensure that existing data resources can be accessed for testing and development of the new CDK based data pipelines.

6. Required items and actions

6.1 AWS Account Creation

There should be two new AWS accounts created; one for dedicated DEV and one for STAGING. The dedicated DEV account will ultimately replace the current DEV environment which is currently shared with CERT. The STAGING environment will provide a "CERT" environment during the transition process, but will ultimately provide a pre-production testing and hot-fix environment.

6.2 Create deployer and cdk-executor policies

On each account both the deployer policy, defined in section 7.1, "Deployer IAM User Policy" and the 7.2, "AWS CDK Execution Policy", needs to be created.

6.3 Create deployer IAM User

On each account the deployer IAM user must be created and have the deployer IAM user policy attached to the user. The user should not have access to the console, and the Deployer IAM Policy is attached directly to the user. Keys should be created for the user, using the create access key link. The use-case for key-creation should be Command Line Interface (CLI). The accesskeys.csv file should be provided to the FPAC for each of these IAM users. There should be 4 of them.

6.4 AWS CDK bootstrap

The AWS CDK bootstrap is a critical one-time action, that is required to enable the AWS CDK. The architecture is utilizing this required process to provide access to the required AWS services. The details of the bootstrap process and installing the AWS CDK are defined in section 7.3, "AWS CDK Bootstrapping". An AWS IAM user with Full Admin must execute the bootstrap process. In this solution, the deployer IAM user cannot perform the bootstrap.

The following accounts and regions must be bootstrapped:

Existing accounts

- PROD us-east-1
- PROD us-east-2
- DEV/CERT us-east-1
- DEV/CERT us-east-2

New Accounts

- DEV us-east-1
- DEV us-east-2
- STAGING us-east-1
- STAGING us-east-2

7. Related Resources

7.1 Deployer IAM User AWS Policy

The IAM user for CI/CD deployments. The preferred name is fpac_cicd_deployer. The FPAC team should receive the csv file of the access keys for this IAM user. This IAM user, with this policy must be created in every AWS account, utilized by FPAC.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCore",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:DeleteChangeSet",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStackResources",
        "cloudformation:ListStacks",
        "cloudformation:ListStackResources",
        "cloudformation:GetTemplate",
        "cloudformation:ValidateTemplate",
        "cloudformation:TagResource",
        "cloudformation:UntagResource",
        "cloudformation:DeleteStack",
        "s3:*",
        "ssm:*"
      ],
      "Resource": "*"
```

```
    },
     {
      "Sid": "AllowPassRoles",
      "Effect": "Allow",
      "Action": [
            "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/cdk-*"
    }
  ]
}
```

7.2 AWS CDK Execution Policy

The document and instructions assume that this policy is named: CDK-EXECUTOR.

This policy has been tested using AWS CDK solutions that mimic the work done by FPAC. This policy was able to deploy, update and destroy all of the artifacts. The preferred name of this policy is CDK-EXECUTOR.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudFormationLimited",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStackResources",
        "cloudformation:ListStacks",
        "cloudformation:ListStackResources",
        "cloudformation:GetTemplate",
        "cloudformation:ValidateTemplate",
```

```
        "cloudformation:TagResource",
        "cloudformation:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowLimitedIAMRoleManagementForCDK",
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:UpdateRole",
        "iam:PutRolePolicy",
        "iam:DeleteRolePolicy"
      ],
      "Resource": "arn:aws:iam::*:role/Fpac*"
    },
    {
      "Sid": "AllowIAMRoleManagementLimited",
      "Effect": "Allow",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:GetRole",
        "iam:PassRole",
        "iam:ListRoles",
        "iam:GetRolePolicy",
        "iam:ListRolePolicies"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowAWSServices",
      "Effect": "Allow",
      "Action": [
        "tag:*",
        "ssm:*",
        "sts:*",
        "s3:*",
```

```
            "lambda:*",
            "states:*",
            "glue:*",
            "dms:*",
            "logs:*",
            "dynamodb:*",
            "athena:*",
            "sns:*",
            "kms:*",
            "sqs:*",
            "airflow:*",
            "airflow-serverless:*",
            "codebuild:*",
            "codepipeline:*",
            "events:*",
            "dsql:*",
            "cloudtrail:*",
            "observabilityadmin:*",
            "applicationinsights:*",
            "evidently:*",
            "synthetics:*",
            "datapipeline:*",
            "databrew:*",
            "ses:*",
            "rds:*",
            "rds-data:*",
            "secretsmanager:*",
            "redshift:*",
            "redshift-serverless:*",
            "redshift-data:*",
            "kafka-cluster:*",
            "kinesis:*",
            "kinesisanalytics:*",
            "apigateway:*",
            "cloudfront:*",
            "acm:*",
            "cognito-idp:*",
            "cognito-identity:*",
            "cognito-sync:*",
```

```
        "route53:ListHostedZones",
        "route53:ListHostedZonesByName",
        "route53:GetHostedZone",
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowReadOnlyEC2Networking",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeRouteTables"
      ],
      "Resource": "*"
    }
  ]
}
```

7.3 AWS CDK Bootstrapping

The deployer IAM user has a very restricted AWS policy, to enable such a restricted policy requires that someone with Full Admin privilege run the CDK bootstrap process. This is the command to run the bootstrap, which will attach the CDK-EXECUTOR policy to the AWS CDK, for the particular account and region. This assumes that the AWS CDK is installed in an environment to perform the bootstrapping. There is further AWS documentation on installing and using the AWS CDK.

```
cdk  bootstrap aws://<account>/<region>  --cloudformation-execution-
policies arn:aws:iam::<account>:policy/CDK-EXECUTOR
```

7.4 Getting Started with AWS CDK

Helpful documentation:

Resource	Link	Description
AWS CDK API Reference (v2)	https://docs.aws.amazon.com/cdk/api/v2/	Official AWS CDK API documentation for all constructs, classes, and modules.
Book: *AWS CDK in Practice*	https://www.amazon.com/gp/product/B0BJF8PRHD/	Practical guide to building real AWS solutions using the CDK with TypeScript and Python.
GitHub Repo for *AWS CDK in Practice*	https://github.com/PacktPublishing/AWS-CDK-in-Practice	Source code examples that accompany the book, including CDK patterns and full project samples.
Getting Started with AWS CDK	https://docs.aws.amazon.com/cdk/v2/guide/getting_started.html	Official AWS quick-start tutorial for learning CDK basics and deploying your first stack.

7.4.1 Installing the AWS CDK (Cloud Development Kit)

This guide walks a new user through installing the AWS CDK, validating the installation, configuring AWS credentials, and preparing the environment for CDK development.

📌 Prerequisites

Before installing the AWS CDK, ensure the following are installed on your machine:

** Node.js (LTS version recommended)**

The AWS CDK is distributed as an NPM package.

Download Node.js:
https://nodejs.org/en/download/

Verify installation:

```
node –v
npm –v
```

Install the AWS CLI

```
npm install –g aws–cdk
```

Installing AWS CDK

Requires NodeJS

```
npm install –g aws–cdk
```

📚 Helpful Resources

Resource	Link	Description
AWS CDK Best Practices	https://docs.aws.amazon.com/cdk/v2/guide/best-practices.html	Architecture, security, and design best practices recommended by AWS for CDK projects.
AWS CDK GitHub Repository	https://github.com/aws/aws-cdk	Official AWS CDK source code, issues, discussions, and examples.