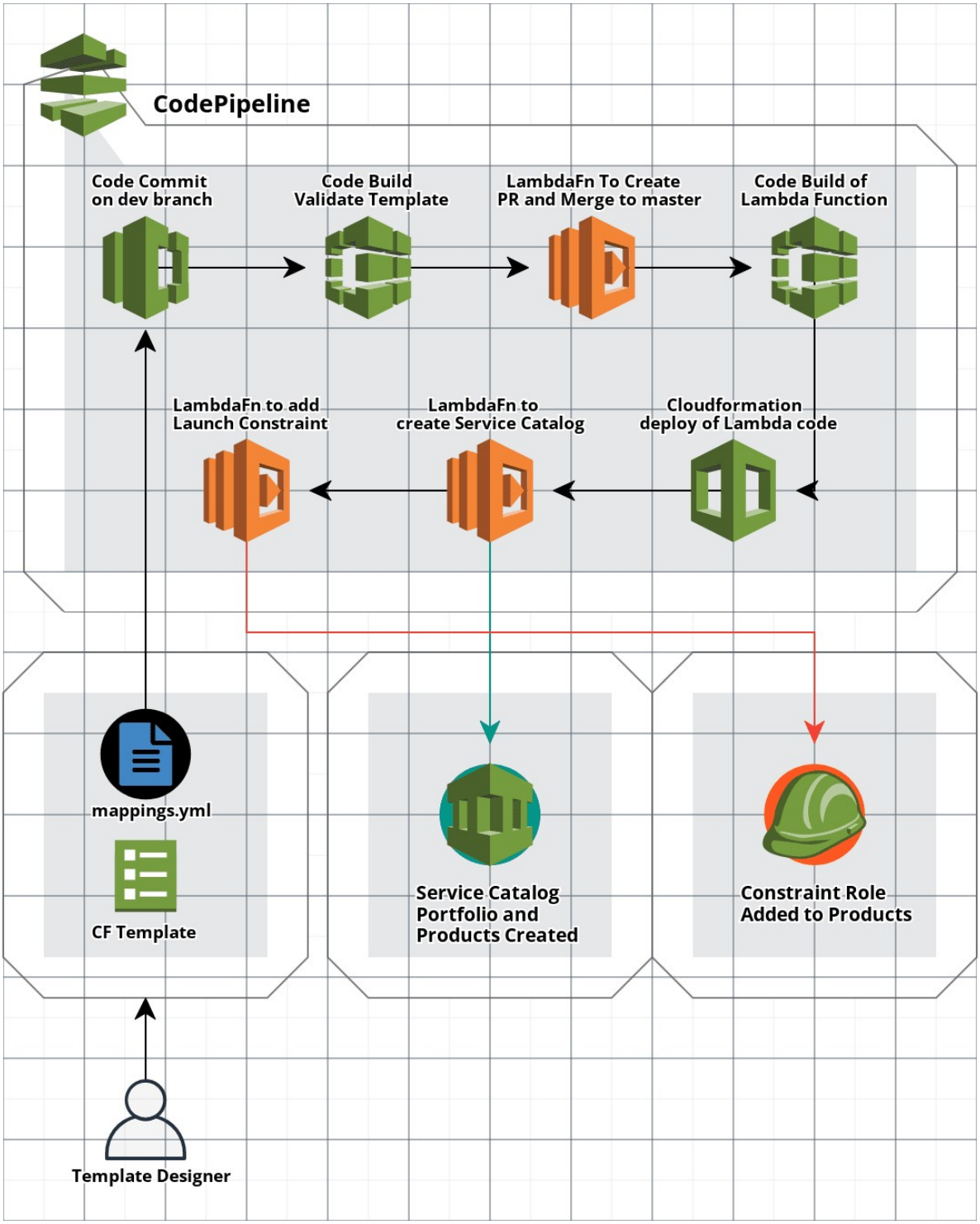
Code Pipeline Deploy of Service Catalog Portfolios and Products



Requirements:

Portfolio Folders:

- All Portfolio folders must begin with "portfolio-".

Products Template:

- All product templates must be Cloudformation templates in YAML or JSON.
- All product templates must start with "product-".

Mappings file:

- File name must be “mapping.yaml”.
- This file and its makeup are key to deploying your portfolio and products.
- mapping.yaml must be in the root of each Portfolio folders.
- mapping.yaml layout.

```
name: portfolio name (Example: portfolio-dev1)
description: Description of portfolio
owner: Owner Name
products:
- name: Name of Product (Example: vpc-product)
template: Product Template File (Example: product-vpc.yaml)
owner: Owner Name
description: Description of Product
accounts:
- identifier: AWS Account
number: AWS Account Number of where produc is being installed
tags:
- Key: key name
Value: Value to associate with key name
principals:
- Group Name to associate with Service Catalog (Example: group/SC-Product-Dev1-UserGroup)
```

This deploys 2 Lambda Functions, CodePipeline, 2 User Groups, Codestar Notification for Pipeline Failure, SNS Topic that Codestar uses to send out an email for notification, and a CodeCommit Repository that triggers the CodePipeline on code committed to the dev branch.

Pipeline Steps:

1. Source Code committed to the dev branch.
2. Code Build Test job to validate the Cloudformation template.
3. Lambda function to create a pull request and merge to the master branch.
4. Code Build job to build and package Lambda function for deploying Service Catalog products.
5. Cloudformation job to deploy and create the Lambda function.
6. Lambda function to deploy the Service Catalogs from artifacts.
7. Lambda function to create Launch Constraint on the Service Catalog products.

Deploy Steps:

Use Runway to deploy in the following order:

- ccmerge.sls
- launch_constraint.sls
- pipeline.cfn

Make sure your branch is set to dev or your DEPLOY_ENVIRONMENT variable is set to dev