AWS CloudFormation Stacks and Automation: Best Practices

BEST PRACTICES FOR MANAGING CLOUDFORMATION TEMPLATES

Ryan Lewis
CLOUD ARCHITECT
@ryanmurakami ryanlewis.dev

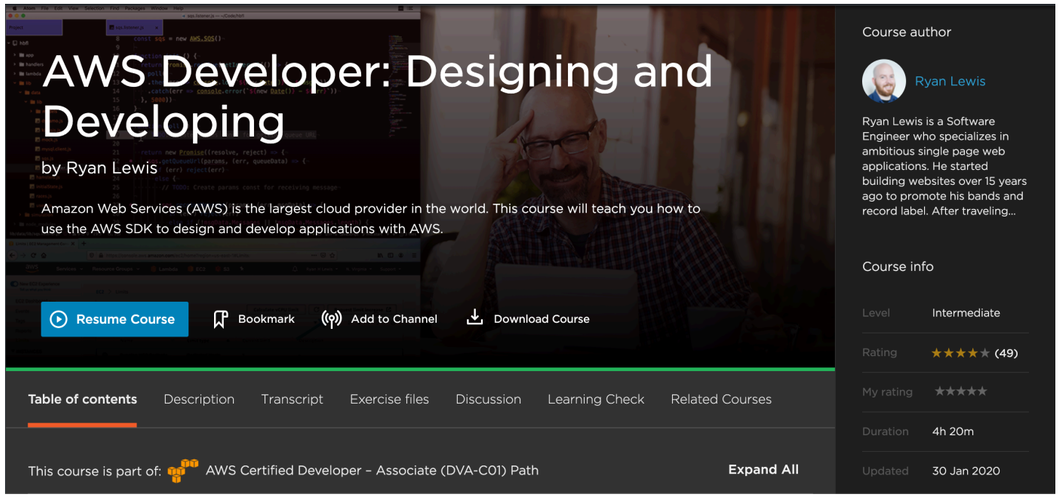
By the End of This Course

Best practices for existing projects

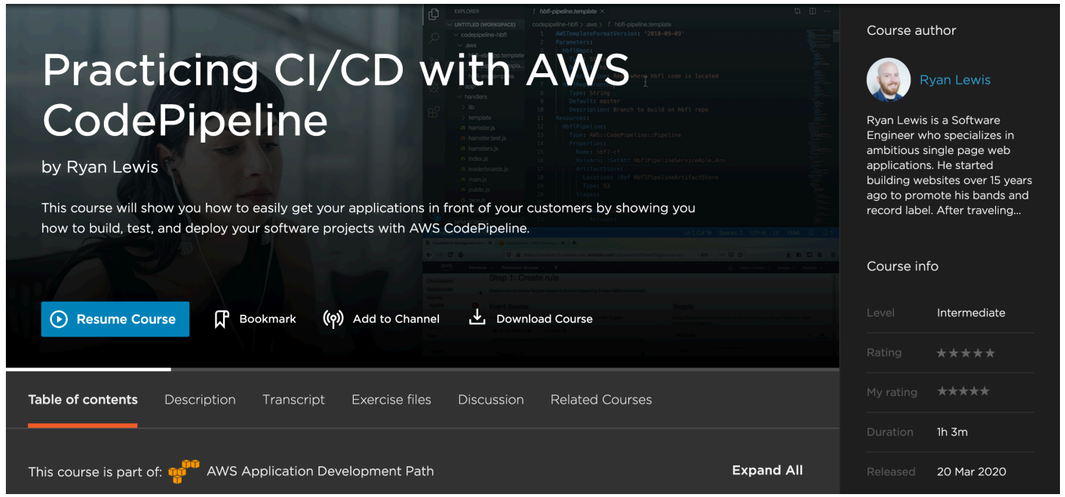
Best practices for new projects

Meet Candace: DevOps Extraordinaire

Previously on Hamster Ball Fantasy League



AWS Developer: Designing and Developing



Practicing CI/CD with AWS CodePipeline

The Hamster Ball Fantasy League Empire

HBFL

Races Hamsters Leaderboards Login

Hamster Ball Fantasy League

The ONLY place on the web to follow the Hamster Ball League

Featured Hamsters

Zepto

Rank:

Type: Speedball

Milkshake

Rank:

Type: Speedball

Fievel

Rank:

Type: Tiny Terror

CloudFormation at HBFL

More

Less

The Search for Best Practices Begins

Module Overview

Best Practices for Managing CloudFormation Templates

Building and Organizing Templates

Module Overview

Best Practices for Managing CloudFormation Stacks

Managing and Optimizing Stacks

Overview

Securing CloudFormation with IAM

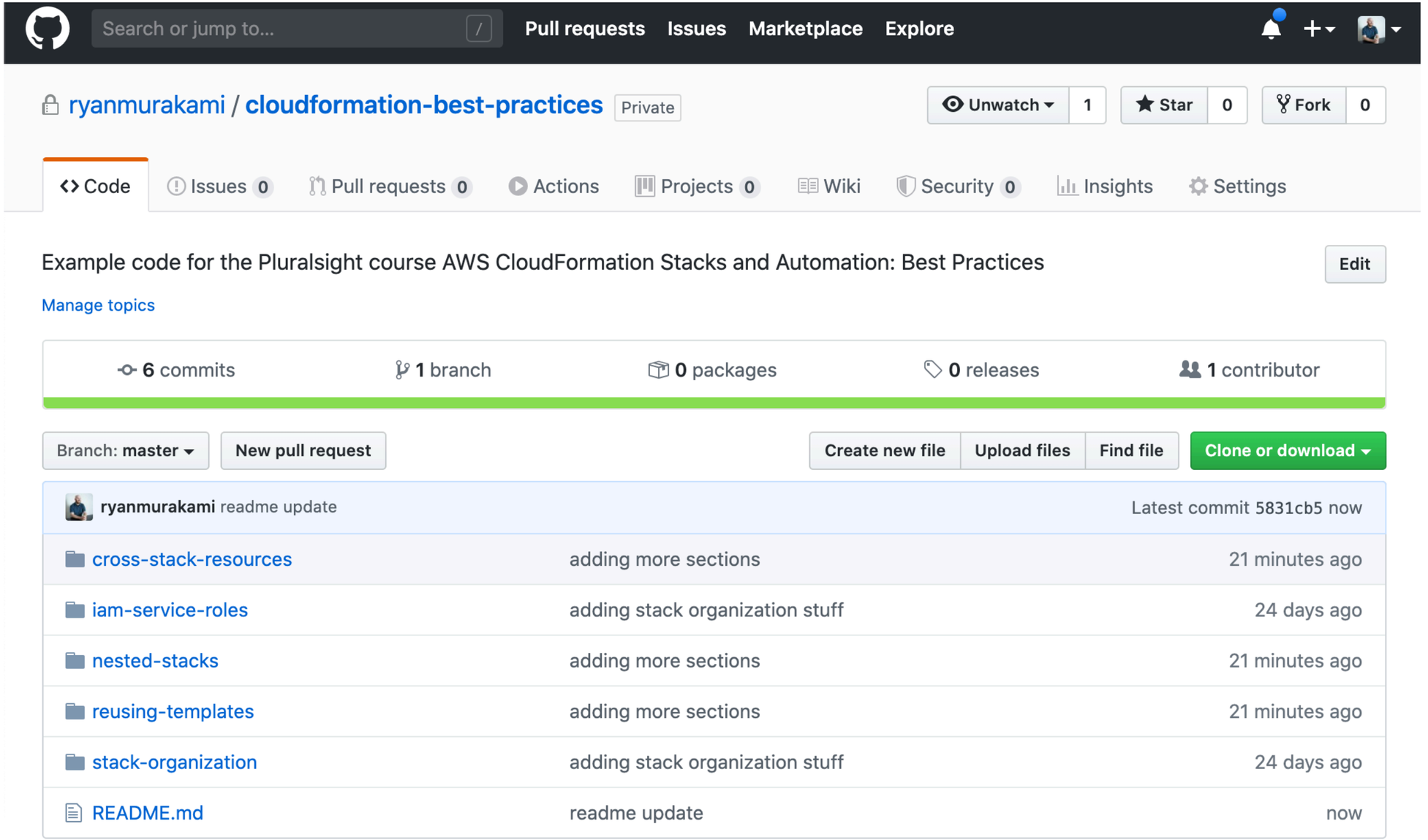
Mind the Service Quota gap

Cleaning up the CloudFormation mess

Out with the single-use templates

Creating matryoshka templates

Output sharing for much win

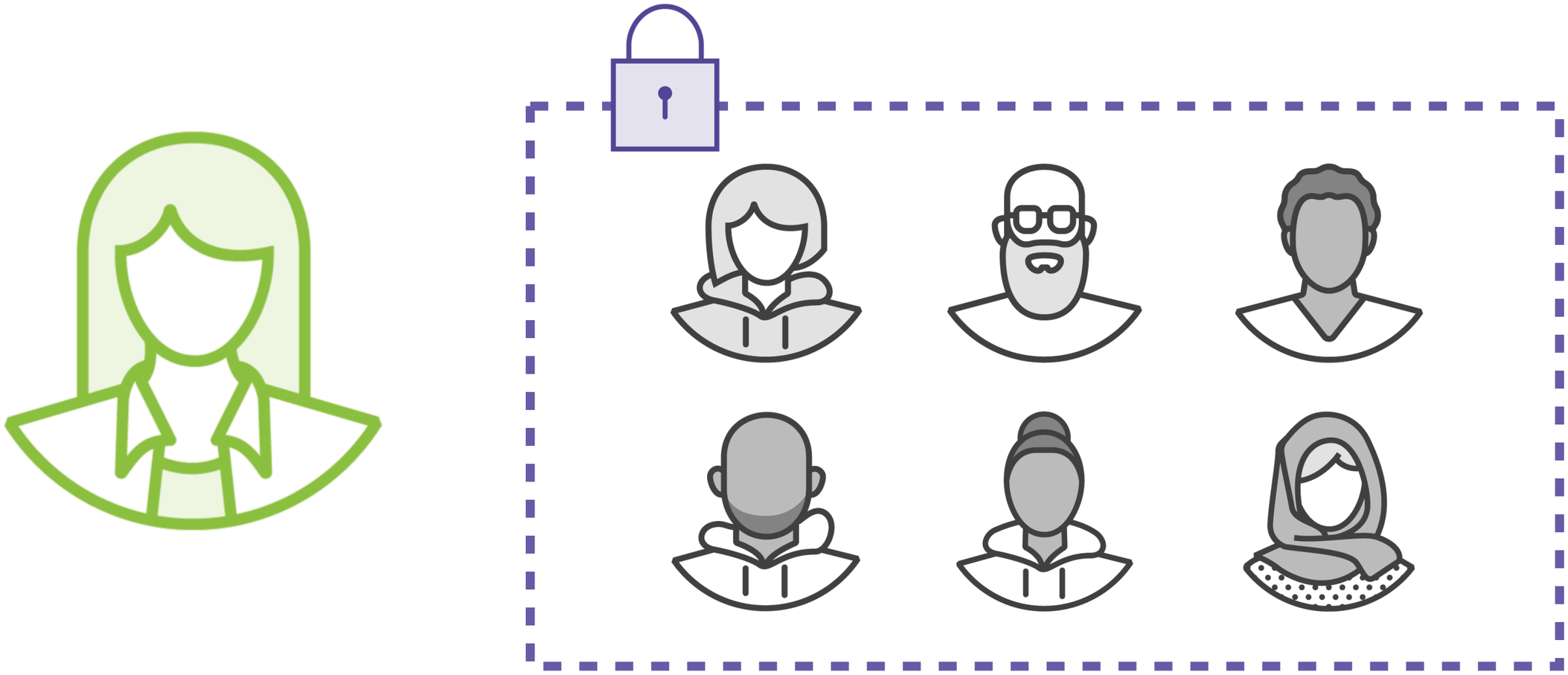


https://github.com/ryanmurakami/cloudformation-best-practices

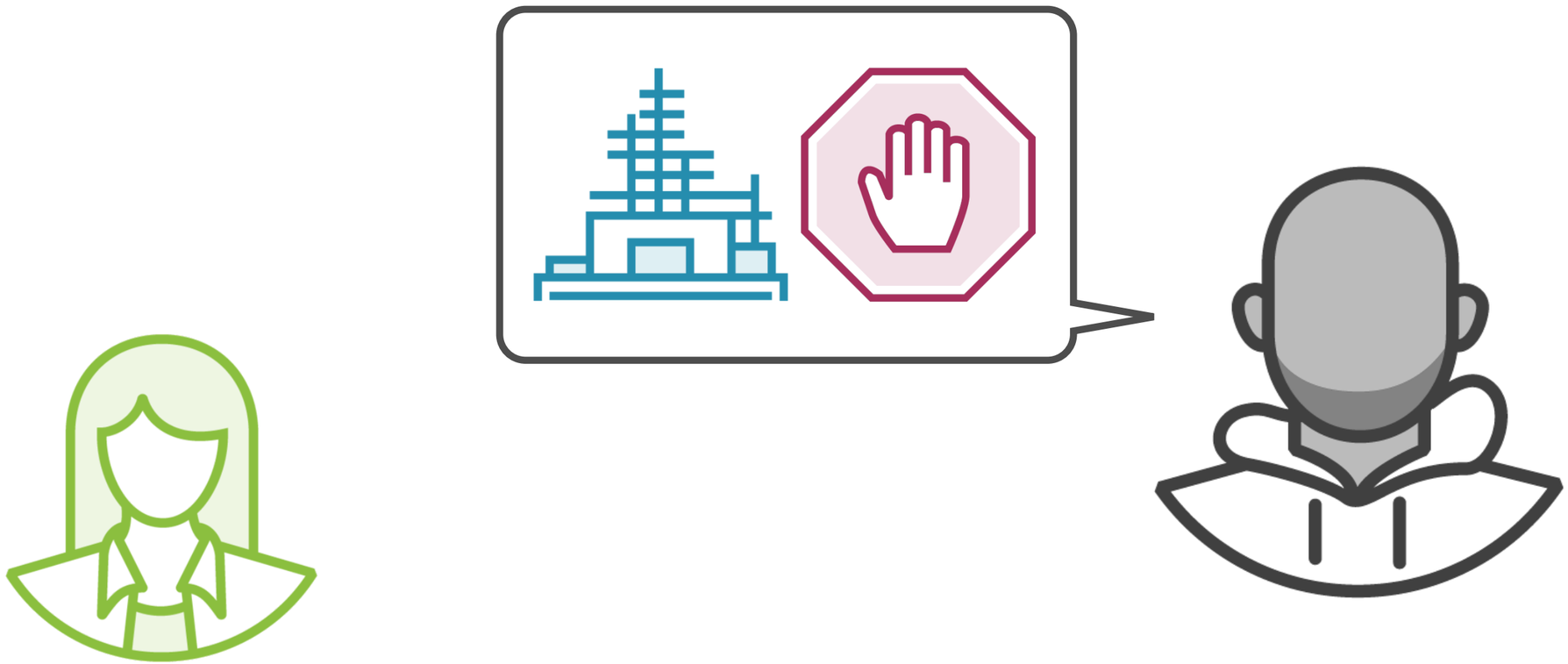
Let’s get best practicing!

Using CloudFormation Securely with IAM

Engineering Security at HBFL



CloudFormation Creation Denied



Should Candace Add Permissions to the User?

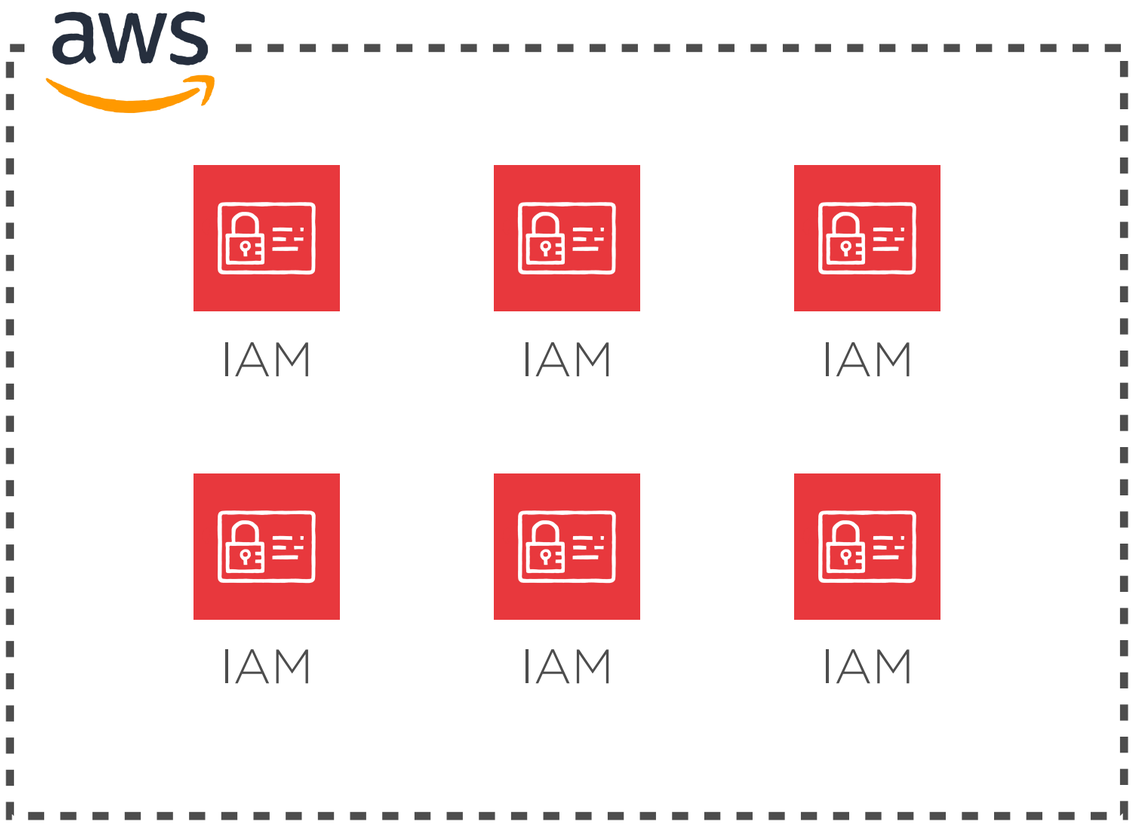
Subverts security protocols

One-off requests for each user

CloudFormation Best Practice

Use IAM to Control Access

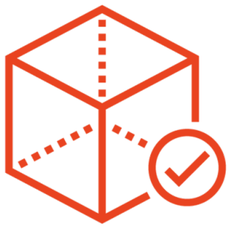
A CloudFormation Template Is Executed


aws
IAM
IAM
IAM
IAM
IAM
IAM

A CloudFormation Template Is Executed

Executing CloudFormation Templates

Permissions of the user who executed it is used



If user can create anything, CloudFormation can create anything

If user is restricted, CloudFormation is restricted

Managing Access in CloudFormation

CloudFormation Service Role

IAM role with CloudFormation trust policy

Attached policies permissions are used instead of user's

Useful when using CloudFormation in CI/CD

Demo

Creating and using service roles

CloudFormation Service Roles at HBFL

Verifying Quotas for CloudFormation Stacks

HBFL AWS Resources Increase

AWS Account Resource Limits

Each resource has its own limits

Each account has different limits

Most limits can be increased

CloudFormation Best Practice

Verify Quotas for All Resource Types

HBFL AWS Service Limits Increase

Organizing Your CloudFormation Stacks and Templates

Team Organization at HBFL

HBFL by the Numbers

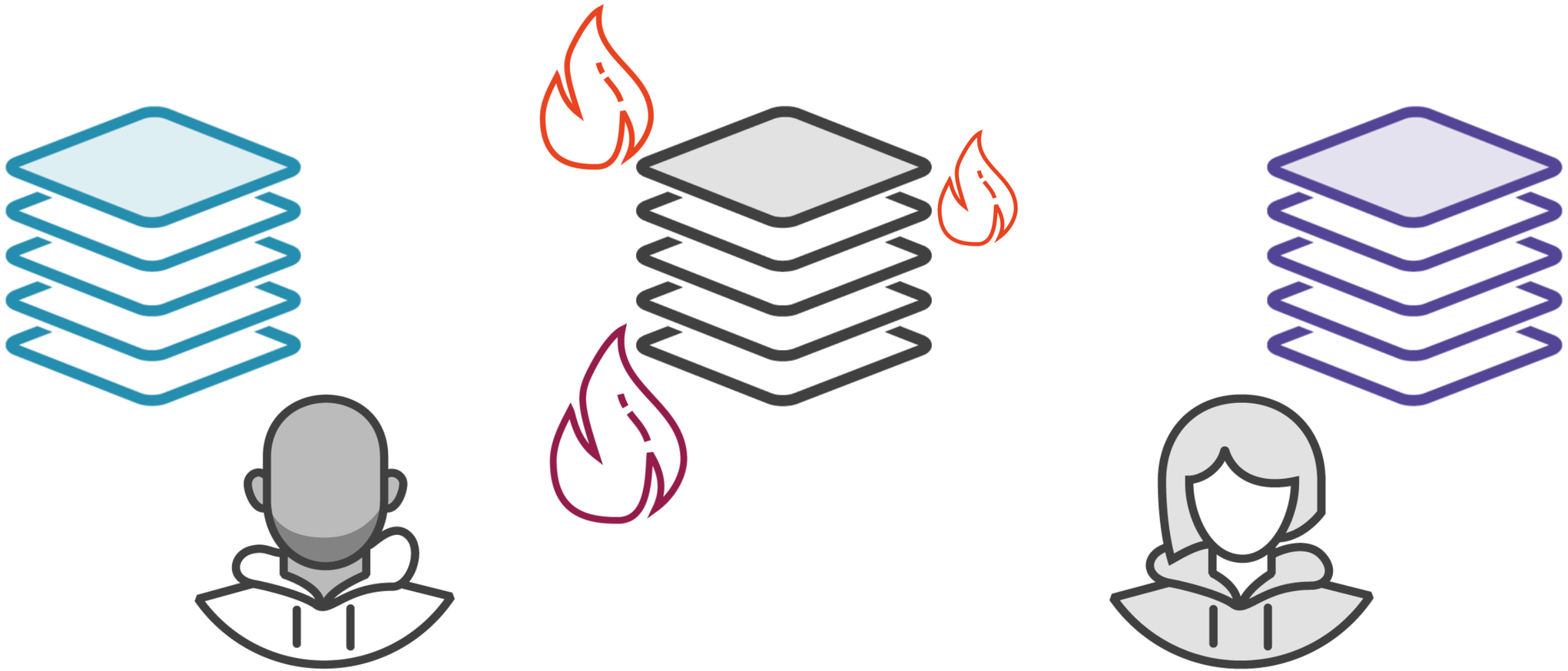
14

Engineering Teams

11

CloudFormation Templates

Stack Conflicts

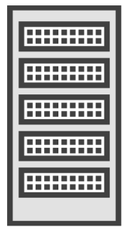


CloudFormation Best Practice

Organize Your Stacks by Lifecycle and Ownership

CloudFormation Stack Organization

Team A

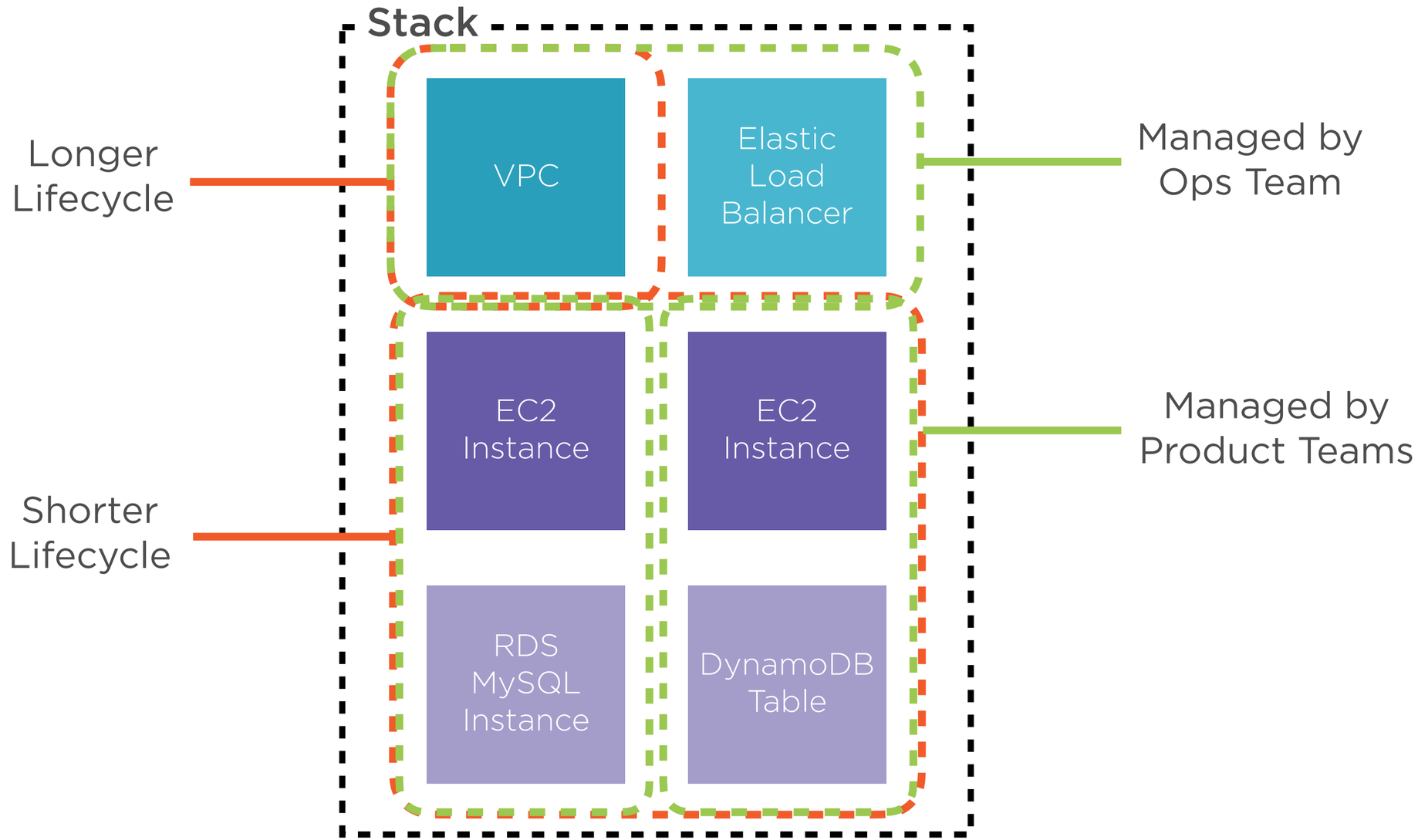


Team B

Team C

Team D

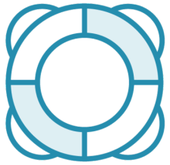
The Mono-stack Enters



Issues arise when **many** teams manage the same stacks and templates

Breaking Apart the Mono-stack

Moving Resources between Stacks



Add “Retain” DeletionPolicy to resource and update stack

Remove resource from template and update stack

Add resource to new/existing template with “Retain” DeletionPolicy

Remove “Retain” DeletionPolicy from resource

Some resource types cannot be moved between stacks

CloudFormation Best Practice

Organize Your Stacks by Lifecycle and Ownership

Reusing CloudFormation Templates

Template Upkeep at HBFL

CloudFormation Best Practice

Reuse Templates to Replicate Stacks in Multiple Environments

When to Reuse Templates

Deploying same resources to different region or environment

Share templates between products and teams

Reuse resources between stacks

Parameters

Real values passed in when template is executed

Useful for instance size, region, names and more

Mappings

Useful for standardizing values based on parameters

Reduces the number of parameters needed

Conditions

Evaluate parameters for conditional creation or properties

Reasons to Use Separate Parameters Files

Can commit parameters to source control

Shorter and simpler CLI commands

External Mappings File Usage Example

MyStack.yaml

```yaml
Mappings:
  'Fn::Transform':
    Name: 'AWS::Include'
    Parameters:
      Location: 's3://<bucketname>/mappings.yaml'
```

External Mappings File Example

mappings.yaml

```yaml
EBPlatformMap:
    dotnet:
      version: dotnet-v4
      SolutionStackName: 64bit Windows Server…
    node:
      version: node-v12
      SolutionStackName: 64bit Amazon Linux…
    python:
      version: python-v3
      SolutionStackName: 64bit Amazon Linux 2…
```

CloudFormation Best Practice

Reuse Templates to Replicate Stacks in Multiple Environments

Using Nested CloudFormation Stacks

Sharing Templates at HBFL

CloudFormation Best Practice

Use Nested Stacks to Reuse Common Template Patterns

Nested Stack

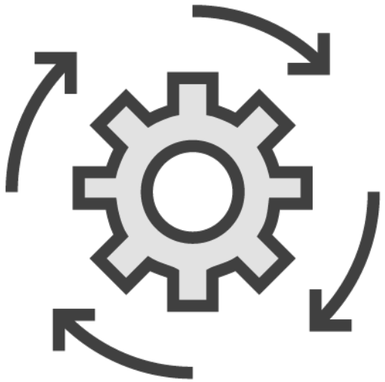
CloudFormation stack defined as a resource inside another template.

Nested Stack Definition

MyStack.yaml

```yaml
Resources:
  MyNestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://s3.amazonaws.com/
                   mybucket/eb.yaml
      Parameters:
        Parameter1: Value1
        Parameter2: Value2
```

Nested Stack Template Execution



When a stack is updated, it will also update any nested stacks by checking the template in the referenced S3 bucket

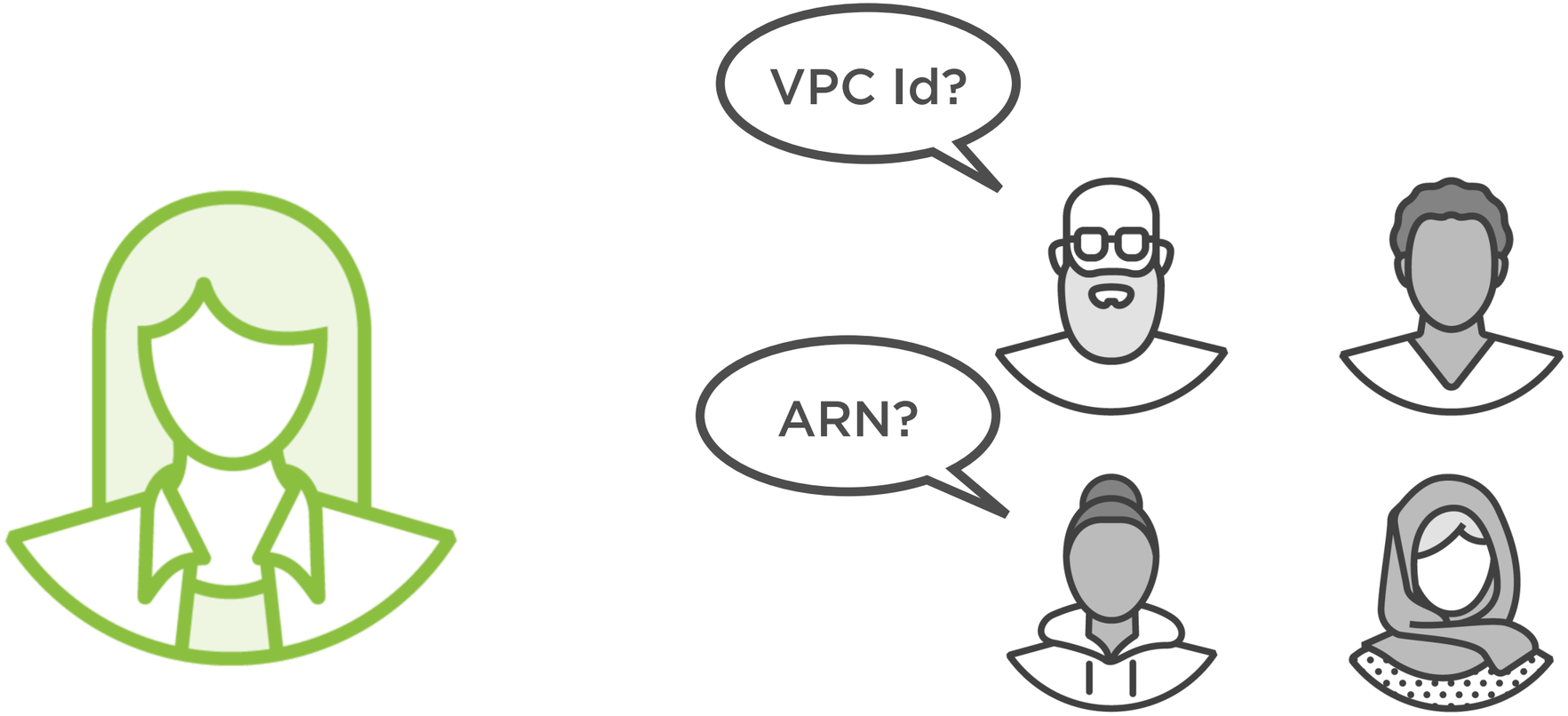
The templates for nested stacks live in S3, so you must reupload the template if you make any changes

A nested stack creation failure will result in the failure of the parent stack as well

Nested Stacks at HBFL

Importing Cross-stack Resources

Sharing Resource Properties at HBFL



CloudFormation Best Practice

Use Cross-stack References to Export Shared Resources

CloudFormation Output

A value from one stack that can be imported into another stack.

Available Output Value Types

Resource ARN

Resource ID

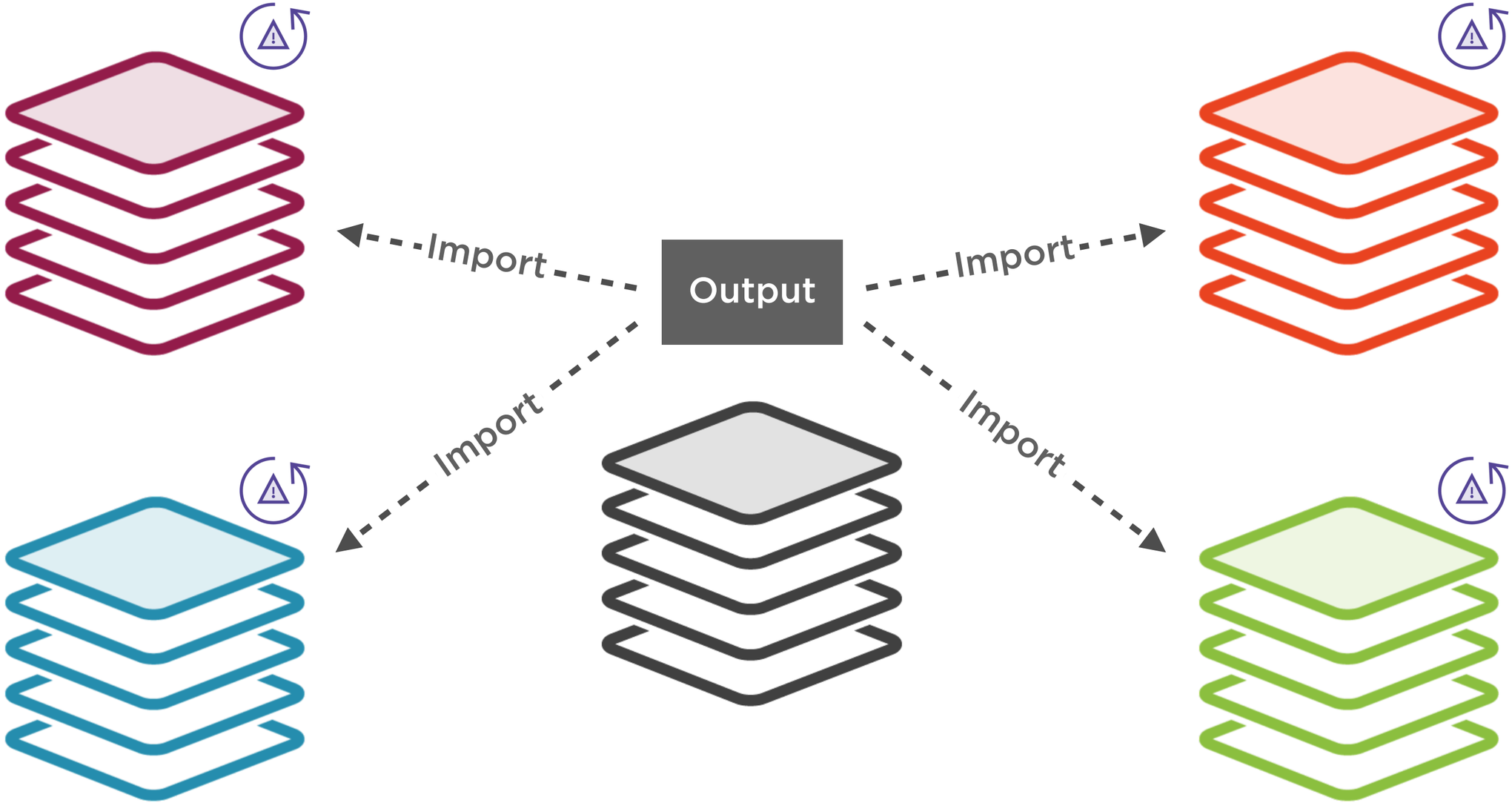
String

Resource Attribute

Use Fn::ImportValue to import cross-stack references

Output values that are imported by other stacks cannot be modified or deleted

Output Referenced by Other Stacks



Listing Imports with the AWS CLI

Terminal

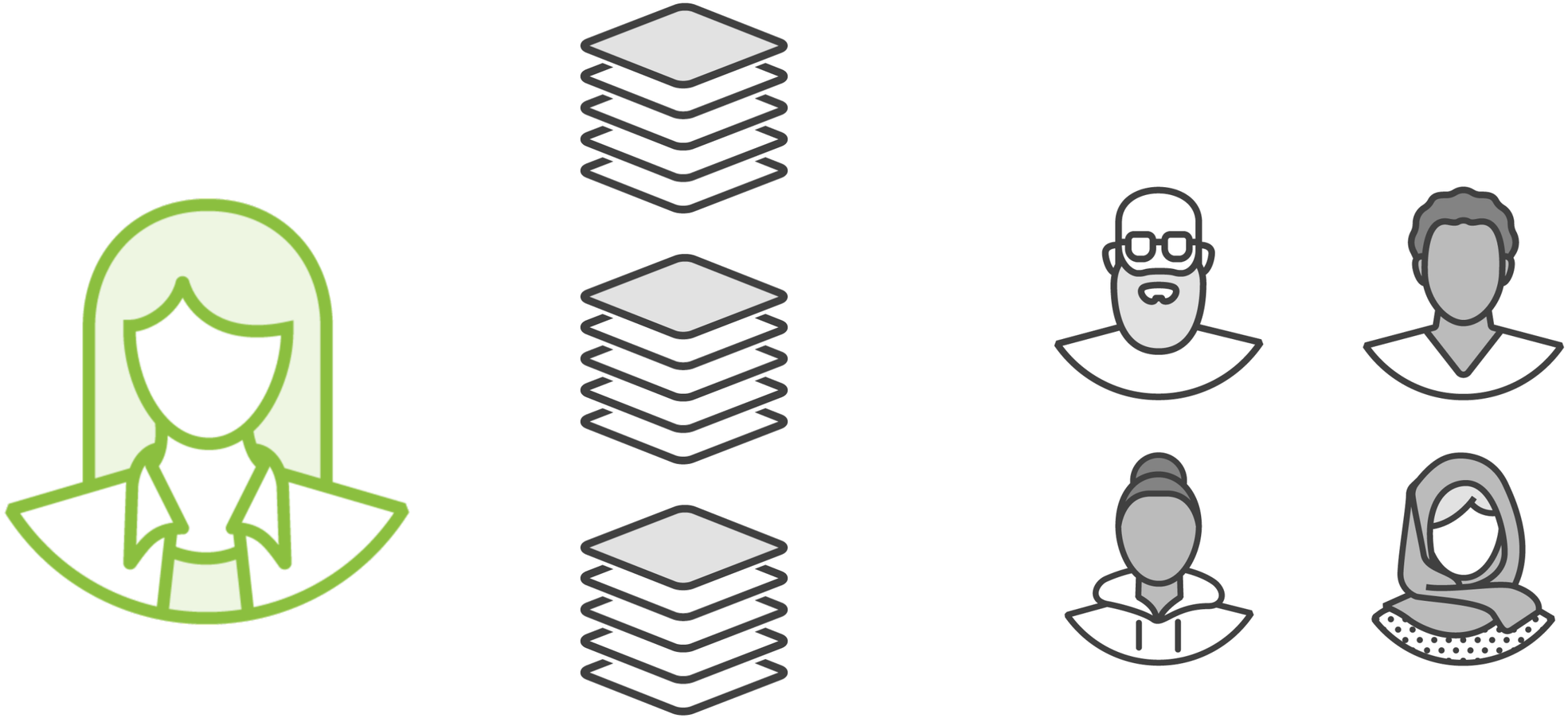
```
> aws cloudformation list-imports --export-name MyExport
```

```
{
    "Imports": [
        “stack-name-that-imports-MyExport“
    ]
}
```

CloudFormation Best Practice

Use Cross-stack References to Export Shared Resources

Cross-stack References at HBFL



Up Next:
Best Practices for Managing CloudFormation Stacks