AWS re:Invent

MGT222-P

Developing resource providers with new AWS CloudFormation tools

Luis Colon

Senior Developer Advocate
AWS CloudFormation
Amazon Web Services

Brian Terry

Senior Partner Solutions Architect
Amazon Web Services

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Agenda

Creating resources: The basics

Coding scenarios

New resource provider development tools

Demo

Motivation

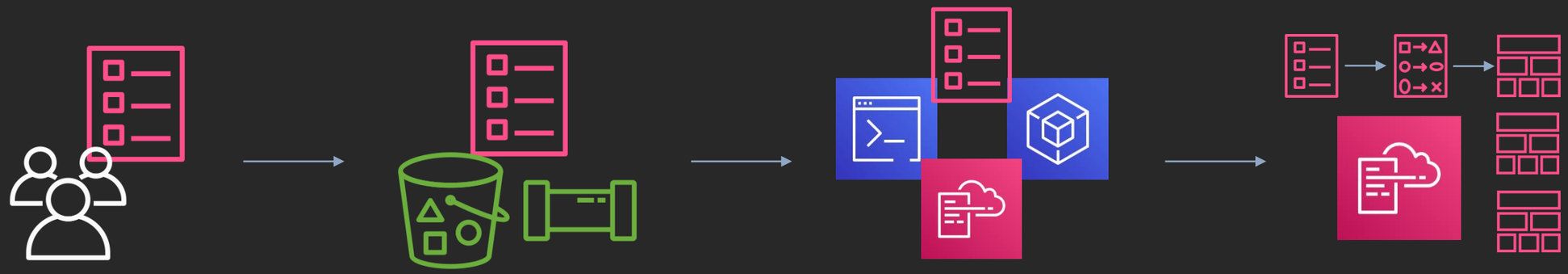
As you leverage more of what AWS has to offer, the number of cloud resources requiring management continues to grow, and application configurations become more complex

Our customers ask us about improving the programmability and customization of many AWS services

They also want to integrate and automate better with their non-AWS resources

Many AWS Management and Governance tools are available to help with enablement, operations and provisioning

Creating resources with AWS CloudFormation



Code

Write templates in YAML, or JSON, or generate code via macros/transforms or higher-level languages (AWS CDK)

Commit

Upload templates directly via browser, from an Amazon S3 bucket, or ideally via CI/CD pipelines to leverage testing tools

Execute

Create stacks from templates using the console, CLI, or AWS SDK, or create stack sets across multiple accounts and regions

Deploy

Stacks, stack sets, and resources are created, updated, and managed in a uniform way

Supports dependencies, rollbacks, stabilization, drift detection, change sets, and more

Coding scenarios in AWS CloudFormation

```
Resources:
  MyServerInstance:
    Type: “AWS::EC2::Instance”
    Properties:
      InstanceType: t2.micro
```



Templates are built in a common language to describe the state of resources

	Core YAML/JSON	Macro/transform	High-level language	Custom resources
Description	Provides for parameters, mappings, outputs, some functions	A mechanism to provide shortcuts and expansions	Write in Python, JS/TS, other languages, convert to Core YAML/JSON	Primarily via AWS Lambda functions in various languages
Notes	Most used; other transformations result in this format	AWS Serverless Application Model (AWS SAM), AWS CloudFormation macros	AWS CDK (multiple languages), Troposphere (Python), others	Call remote APIs, resources not supported yet, proxies to external resources

Custom resources

```
Resources:
  AMIInfo:
    Type: “Custom::AMIInfo”
    Properties:
      ServiceToken: !GetAtt AMIFn.Arn
```

- Usage
  - AWS Lambda (primarily) or Amazon SNS backed
  - Create, update, and delete handlers
  - Most validators (cfn-lint) will not recognize them without customizations
  - Custom::MyTypeName
  - In comparison to native resources, can be cumbersome to create and invoke
  - Widely used
  - Many public and third-party custom resources are available via GitHub

New resource provider tools

```
Resources:
  MyResource:
    Type: “AnyCompany::Service::Resource”
    Properties:
      AnyProperty: 1000
```

- First-class, native experience
  - Invocations are exactly the same as with native resources
  - Resources created with the toolkit can inherit other current features and future enhancements
    - Rollback
    - Change sets
    - AWS CDK
    - And more soon

Components

Core library

- Designed to be language-agnostic
- Java, Python, and Go support at launch

cfn-cli

- Command line tool
- Quickly scaffolds a code project
- Helps with code generation
- Local testing
- Validation
- Submit code

Registry service

- At launch it enables resource providers to be privately shared across users in your account; other sharing models are planned

Open source

```
Resources:
  MyService:
    Type: “Vendor::Service::Tool”
    Properties:
      AnyProperty: 1000
```

- Embracing the community
  - cfn-cli tool will be open-sourced
  - We plan to open source most of our native resource provider code, with several resource providers available at launch
  - We plan to encourage our community on GitHub to implement and share resource providers

Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Summary

The new resource provider tooling provides coder-friendly automation as you extend AWS CloudFormation to manage AWS and non-AWS resources

Leverage CloudFormation’s unique benefits when creating resources across many scenarios, including management and governance use cases

All customers and developers benefit from open-source contributions for both resource providers and tooling components

Next Steps

AWS CloudFormation Slack Channel:

Twitter DM to @luiscolon1 or @thedanblanco with your email

AWS CloudFormation Projects on GitHub:

https://github.com/aws-cloudformation

AWS CloudFormation:

https://aws.amazon.com/cloudformation/

AWS Management and Governance:

https://aws.amazon.com/products/management-tools/

Thank you!

Luis Colon
Senior Developer Advocate
AWS CloudFormation
licolon@amazon.com

Brian Terry
Senior Partner Solutions Architect
AWS
brianter@amazon.com

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.