
AWS
re:Invent

DOP402

Build your first AWS CloudFormation resource provider

Craig Lefkowitz

Senior Developer Advocate, AWS CloudFormation

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Introduction

The components

Development patterns

Operational patterns

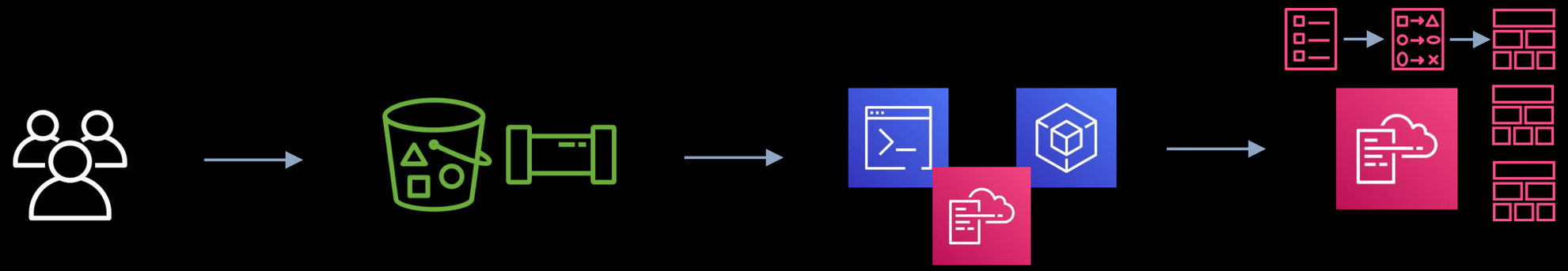
Demo



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Creating resources with AWS CloudFormation



Code

Write templates in YAML or JSON or generate code via macros / transforms or higher-level languages (AWS Cloud Development Kit [CDK])

Commit

Upload templates directly via browser, from an S3 bucket, or ideally via continuous integration and delivery (CI / CD) pipelines to leverage testing tools

Execute

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

Deploy

Create, update, and manage stacks, stack sets, and resources 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	Resource types
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	Write in Go, Python, Java using the registry and CloudFormation CLI
Examples	CloudFormation template	AWS Serverless Application Model (AWS SAM), AWS CloudFormation macros	AWS CDK (multiple languages), Troposphere (Python)	MongoDB, New Relic, Dynatrace

Resource type features

```
Resources:
  MyResource:
    Type: YourCo::Service::Resource
    Properties:
      AnyProperty: 1000
```

- First-class objects
- AWS services integration
- Registry
- Configurable timeouts
- Direct response to CloudFormation

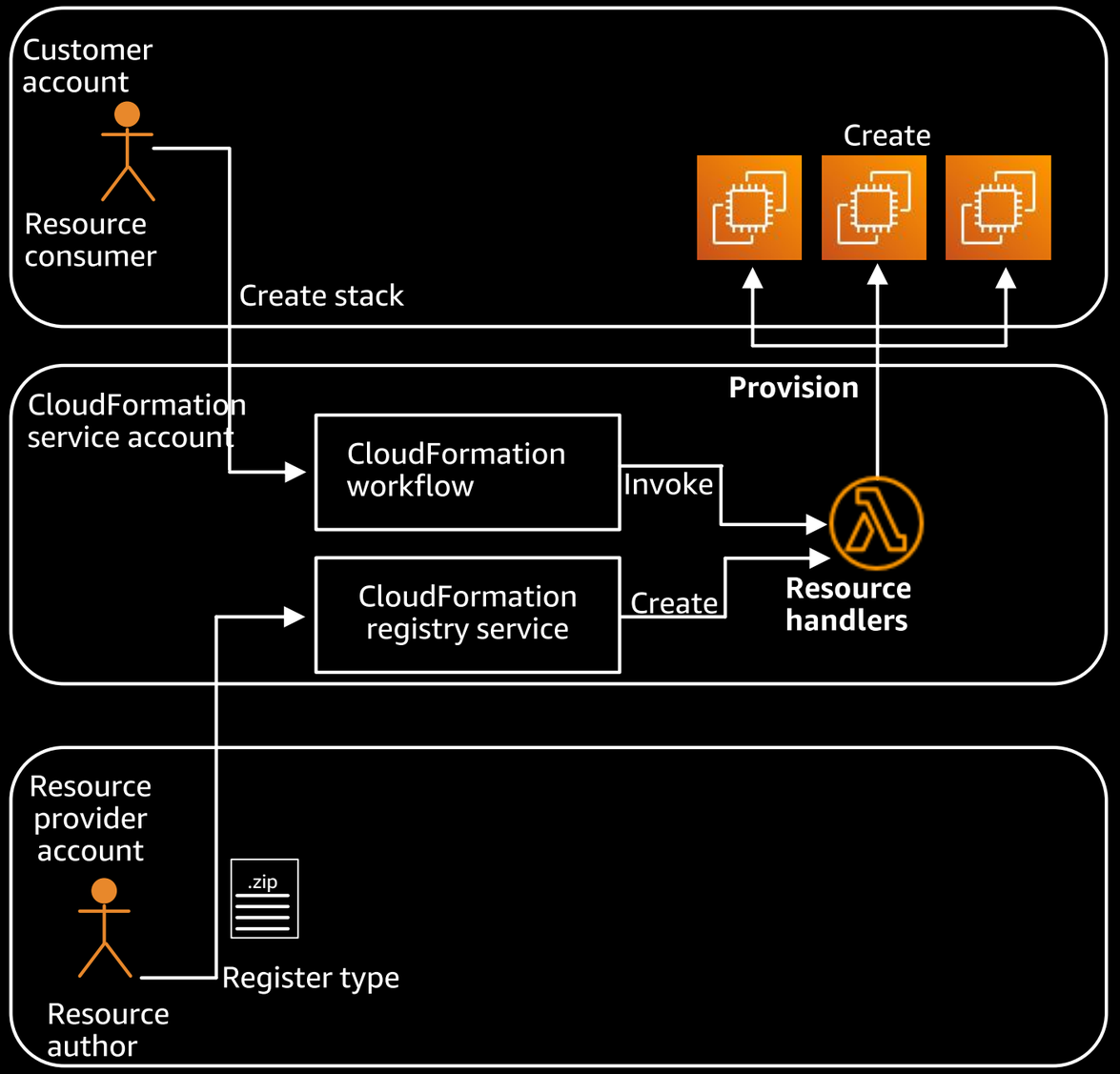
Resource type execution

Resource Lifecycle:

- Author resource type (code, test, build, package)
- Submit resource type to the CloudFormation registry
- Use resource type in a CloudFormation template

Notable Differences:

- Handler code executed in CloudFormation service account
- Schema first design (strong typing)
- Configurable timeout
- CloudFormation CLI
- CloudFormation registry



Components

Registry service

- Enables resource types to be privately shared across users in your account; other sharing models are planned

cfn-cli

- Command line tool
- Project scaffolding
- Code generation
- Local testing
- Validation
- Registry submit

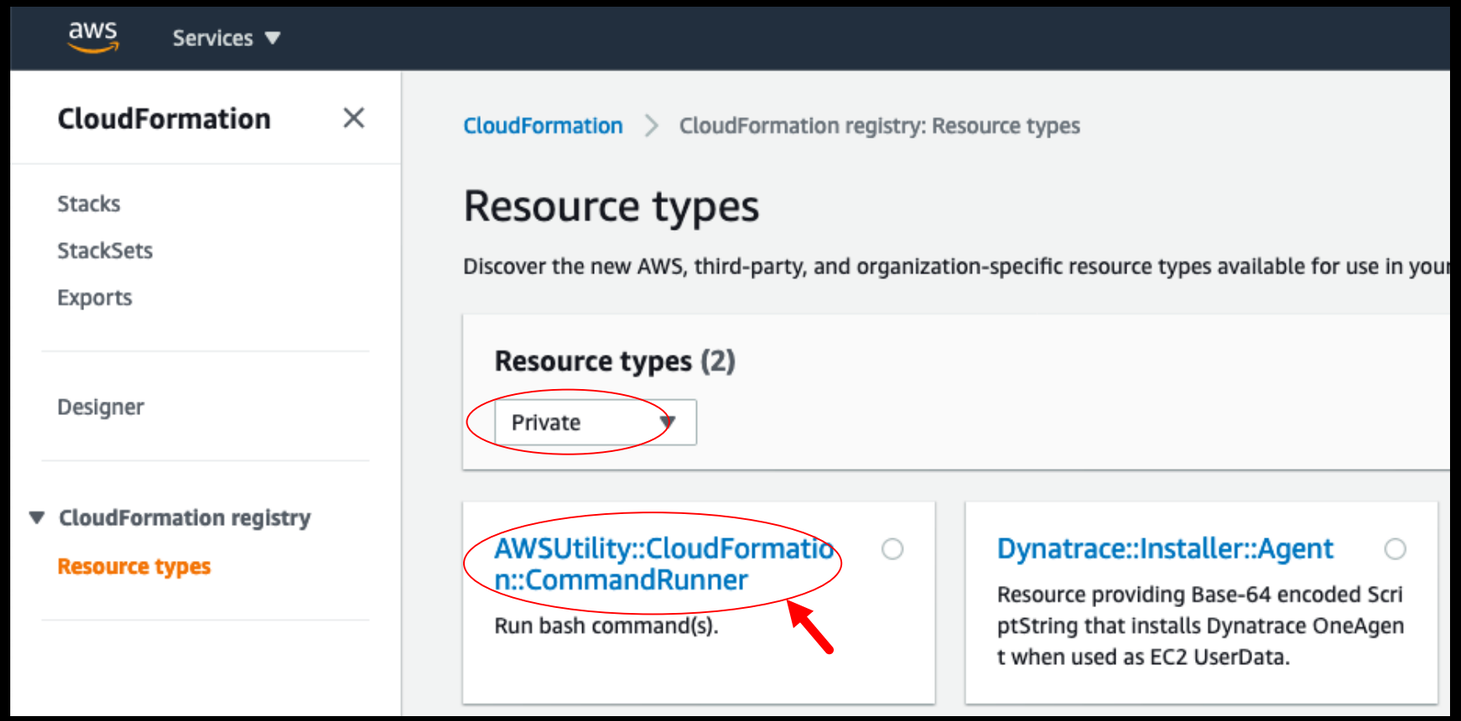


Core library

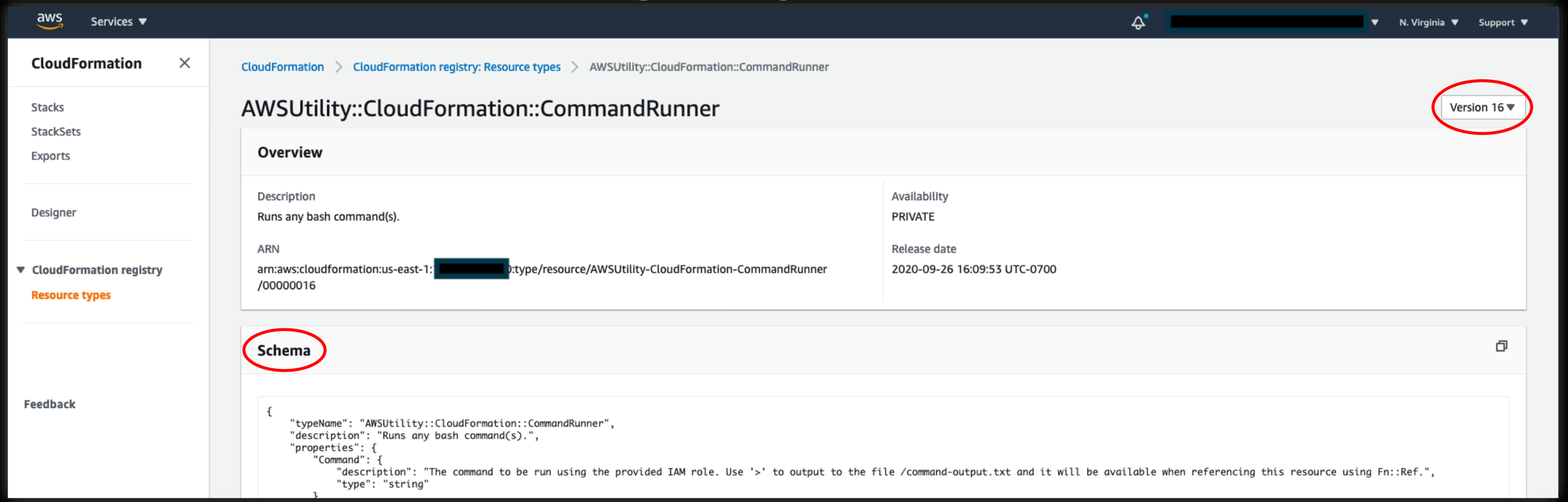
- Designed to be language-agnostic
- Java, Python, and Go

CloudFormation registry

- Centralized versioned repository
- Resource types
- Public and private sections



CloudFormation registry



AWS CLI:

```
aws cloudformation describe-type --type RESOURCE --type-name DemoCo::Service::Resource
```

The CloudFormation CLI

```
> pip install cloudformation-cli
➢ pip install cloudformation-cli-python-plugin

> cfn init

> cfn generate

> cfn test

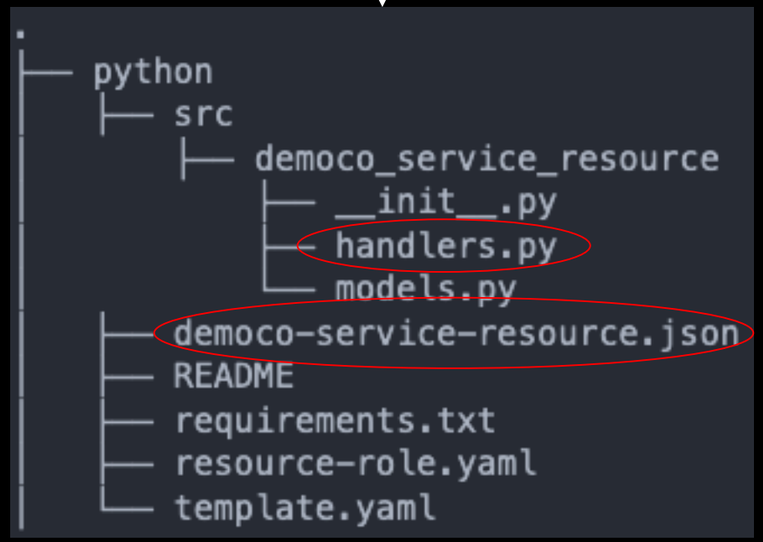
> cfn submit
```

- Install via pypi
- Initialize
- Test
- Submit

A typical development pattern

DemoCo::Service::Resource

```
> mkdir demo-resource
> cfn init
```



Define your model
`democo-service-resource.json`

Author handler code

Python: `src/handler.py`
Go: `cmd/resource/resource.go`
Java: `src/demo/service/resource/*.java`

Build, test, submit

```
> cfn generate
> cfn test
> cfn submit --set-default
```

Test your resource in CloudFormation

Resource type schema – Model your resource

awsutility-cloudformation-commandrunner.json

```
{
  "typeName": "AWSUtility::CloudFormation::CommandRunner",
  "description": "Runs any bash command(s).",
  "properties": {
    "Command": {
      "description": "The command to be run.",
      "type": "string"
    }
  },
  "readOnlyProperties": [
    "/properties/Output",
    "/properties/Id"
    ],
"handlers": {
  "create": {
    "permissions": [
      "ec2:DescribeSubnets"
```

Inputs: `properties`

Outputs/Attributes (`!GetAtt`) : `readOnlyProperties`

Handlers: `CREATE, UPDATE, DELETE, READ, LIST`

Permissions

Resource type schema (resource permissions)

AWSUtility::CloudFormation::CommandRunner

```
"handlers": {
  "create": {
    "permissions": [
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcs",
      "ec2:DescribeSecurityGroups”
      ]
  },
  "read": {
    "permissions": [
      "ssm:GetParameter",
      "cloudwatch:PutMetricData"
      ]
  },
  "delete": {
    "permissions": [
      "ssm:DeleteParameter"
      ]
  }
```

- Defined per CloudFormation action

```
> cfn generate
```

resource-role.yaml



Resource types – Handlers

Five handlers support the COMPLETE lifecycle of a resource

Single entry point

Two additional handlers

- CREATE
- UPDATE
- DELETE

READ

!GetAtt (Fn::GetAtt)

Resource imports

Drift detection

- READ
- LIST



LIST

Future use cases involving resource imports

CloudFormation API to list resources of a specific type

Resource type handler basics

```
_handler(
...

return ProgressEvent(
OperationStatus=“SUCCESS”,
ResourceModel={...}
Message=“Complete”,
)
```

- Each handler has a single entry point
- Handlers must not leak resources
- Handler returns a ProgressEvent object within 60 seconds
(30 seconds for READ/LIST)

Q: What if my handler takes longer than 60 seconds?
A: Use callbacks (asynchronous execution)

ProgressEvent

```
_handler(
...

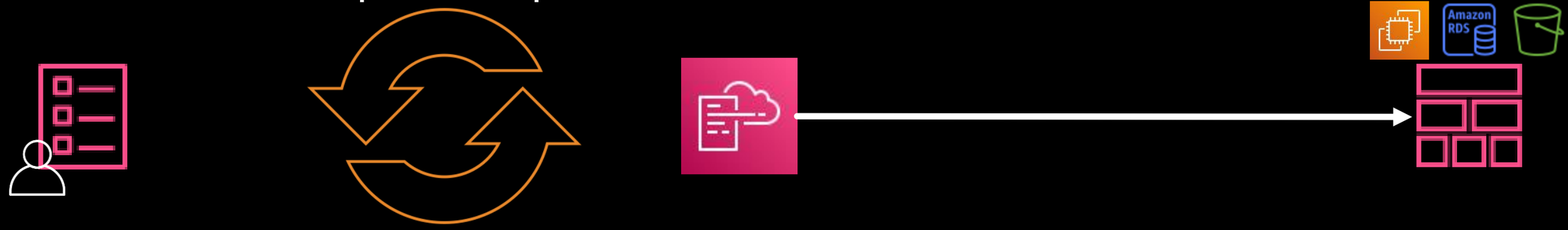
return ProgressEvent(
OperationStatus=“SUCCESS”,
ResourceModel={...}
Message=“Complete”,
)
```

- A JSON object returned from the handler
- Describes current status of the resource
  - SUCCESS
  - FAILED
  - IN_PROGRESS
- OperationStatus SUCCESS
  - Return ResourceModel
- OperationStatus FAILED
  - Return HandlerErrorCode
    - InternalError
    - InvalidRequest
    - Throttling

Best Practice:
Provide a descriptive message for FAILED or IN_PROGRESS Operation Status

Resource type callback

CREATE | UPDATE | DELETE



timeoutInMinutes

```
_handler(
...

return ProgressEvent(
 OperationStatus =“IN_PROGRESS”,
 CallbackDelaySeconds=“300”,
 CallbackContext={...},
 Message=“Processing”
)
```

```
_handler(
...

return ProgressEvent(
 OperationStatus=“SUCCESS”,
 ResourceModel={...},
 Message=“Complete”,
)
```

Error handling and logging



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Error handling

```
_handler(
...

return ProgressEvent(
OperationStatus="FAILED",
HandlerErrorCode="InvalidRequest"
Message="400 Bad Request"
)
```

Handler Error Codes:

- InternalError (terminal)
- NetworkFailure (retriable)
- Throttling (retriable)

Timestamp	Logical ID	Status	Status reason
2020-07-06 13:46:36 UTC-0700	unicorn	DELETE_COMPLETE	-
2020-07-06 13:46:33 UTC-0700	unicorn	DELETE_IN_PROGRESS	User Initiated
2020-07-06 13:44:34 UTC-0700	unicorn	ROLLBACK_COMPLETE	-
2020-07-06 13:44:33 UTC-0700	Unicorn	DELETE_COMPLETE	-
2020-07-06 13:44:03 UTC-0700	unicorn	ROLLBACK_IN_PROGRESS	The following resource(s) failed to create: [Unicorn]. . Rollback requested by user.
2020-07-06 13:44:02 UTC-0700	Unicorn	CREATE_FAILED	400 Bad Request
2020-07-06 13:43:59 UTC-0700	Unicorn	CREATE_IN_PROGRESS	-
2020-07-06 13:43:54 UTC-0700	unicorn	CREATE_IN_PROGRESS	User Initiated

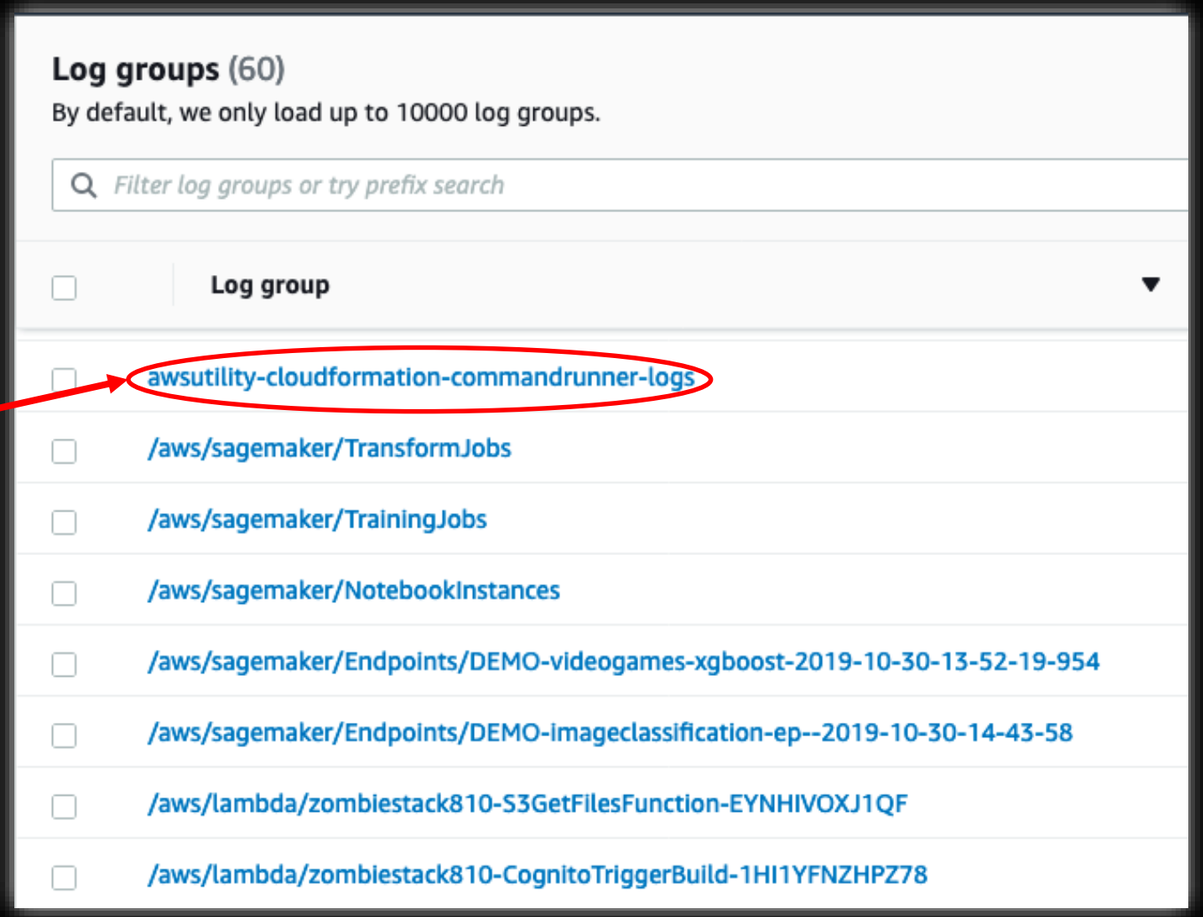
Amazon CloudWatch

CloudWatch Logs Log Group

`AWSUtility::CloudFormation::CommandRunner`

{namespace}-{Service Name}-{Resource Name}-logs

`awsutility-cloudformation-commandrunner-logs`



cfn submit: logging on by default

aws cloudformation register-type: use --logging-config

Resource provider examples



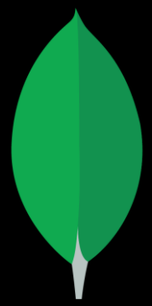
AWS

- SSM
- KMS
- EC2
- IoT
- And more!



Helm

- AWS QuickStart
- EKS
- Charts
- Values



MongoDB

- Projects
- Clusters
- Database users
- IP access lists
- VPC peerings
- Snapshots
- Restores



New Relic

- Alerts



Dynatrace

- Agents

Thank you!

Craig Lefkowitz

Senior Developer Advocate,
AWS CloudFormation
AWS

@CraigLefkowitz

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.