

---

# **AWS Conduit**

*Release 0.0.1*

**May 21, 2018**



<b>1</b>	<b>conduitspec.yaml</b>	<b>3</b>
1.1	Overview . . . . .	3
1.2	Structure . . . . .	3
1.3	Examples . . . . .	3
1.3.1	Serverless Framework . . . . .	3
1.3.2	Service Catalog . . . . .	4
1.4	FAQ . . . . .	4
<b>2</b>	<b>aws_conduit</b>	<b>5</b>
2.1	aws_conduit package . . . . .	5
2.1.1	Submodules . . . . .	5
2.1.2	aws_conduit.conduit module . . . . .	5
2.1.3	aws_conduit.conduit_factory module . . . . .	5
2.1.4	aws_conduit.conduit_portfolio module . . . . .	5
2.1.5	aws_conduit.conduit_product module . . . . .	5
2.1.6	aws_conduit.conduit_s3 module . . . . .	5
2.1.7	aws_conduit.conduit_start module . . . . .	5
2.1.8	aws_conduit.helper module . . . . .	5
2.1.9	Module contents . . . . .	5



Product management for AWS Service Catalog.



**Conduit** [ noun ] - a pipe or tube for conveying water or other fluid.

## 1.1 Overview

The `conduitspec.yaml` is the CI pipeline specification file, which provides metadata about your Cloudformation project to facilitate building, versioning and deploying your application.

The root element of the yaml is the **inventory**, which is a collection of deployable artefacts within your project. Each artefact that is specified in the inventory should be uniquely identified by its **portfolio\*** and **\*\*product** names. The terms portfolio and product have been stolen from AWS Service Catalog, but accurately represent the underlying goal of the `conduitspec.yaml` - the provisioning of a collection of cloudformation templates (products) for a given project (portfolio).

## 1.2 Structure

## 1.3 Examples

### 1.3.1 Serverless Framework

```
inventory:
  - portfolio: "example-portfolio"
    product: "groovy-product"
    sls: True
    build:
      - sls package -s ${STAGE}
    artifact: .serverless/cloudformation-template-update-stack.json
    associatedResources:
      - .serverless/lambdas.zip
```

## 1.3.2 Service Catalog

```
inventory:
  - product: "groovy-product"
    portfolio: "example-portfolio"
    serviceCatalog: True
    artifact: "root-cfn-stack.yml"
    type: "yaml"
    nestedStacks:
      - "first-cfn-nested-stack.yml"
      - "second-cfn-nested-stack.yml"
    associatedResources:
      - "lambdas.zip"
    build:
      - zip lambdas.zip lambdas/*.py
```

## 1.4 FAQ

My pipeline is failing with the error message: “Please do not use wildcard actions in your conduitspec.yaml deployProfile.” We can’t know up front what resources will be provisioned by a CI Pipeline, so the only appropriate IAM role to assign up front is Administrator. That’s a bad idea! The role of the deployProfile is to limit what a CI pipeline can do to only what it needs to do - you as the developer know what resources you are creating / updating / deleting and so you need to tell the CI Pipeline what IAM actions it needs access to.

Assigning a wildcard to an action violates the core principal behind the deployProfile, and so the CI Pipeline will fail if it finds one.

My pipeline is failing with the error message: “Please do not use wildcard resources in your conduitspec.yaml deployProfile.” Same as above but at resource level. Try to be as specific as possible about what resources you are creating.

Whats the HTTP URL for my nested stack? Serverless Framework - This will be generated by Serverless and the CI Pipeline will take care of the rest. Ensure that you have the sls property set to True on your conduitspec product.

Plain Cloudformation - The CI Pipeline will perform some magic to replace instances of \_\_resources\_\_ with the correct S3 bucket and path. All you need is “[https://s3.amazonaws.com/\\_\\_resources\\_\\_/path/to/nested-stack.yml](https://s3.amazonaws.com/__resources__/path/to/nested-stack.yml)”

**MasterStack:** Type: ‘AWS::CloudFormation::Stack’ Properties:

TemplateURL: “[https://s3.amazonaws.com/\\_\\_resources\\_\\_/output/stacks/python/master-python-pipeline.yml](https://s3.amazonaws.com/__resources__/output/stacks/python/master-python-pipeline.yml)” Parameters:

ProjectName: !Ref ProjectName ServiceName: !Ref ServiceName



## **2.1 aws\_conduit package**

### **2.1.1 Submodules**

### **2.1.2 aws\_conduit.conduit module**

### **2.1.3 aws\_conduit.conduit\_factory module**

### **2.1.4 aws\_conduit.conduit\_portfolio module**

### **2.1.5 aws\_conduit.conduit\_product module**

### **2.1.6 aws\_conduit.conduit\_s3 module**

### **2.1.7 aws\_conduit.conduit\_start module**

### **2.1.8 aws\_conduit.helper module**

### **2.1.9 Module contents**