Validating Templates

Get ready to learn how to validate the template files which configure your application!

WE'LL COVER THE FOLLOWING

- CFN Lint
  - Yamllint and ESlint
  - SAM validation

YAML makes it easy to read CloudFormation template files but is very fiddly and error-prone. Small spacing issues can cause problems and that might not be immediately obvious when you read a template. CloudFormation will try deploying resources even when the template is not fully valid and then fail when it starts processing an invalid resource, so feedback on errors might take a long time with complex templates. One of the projects I work on includes custom DNS setup for a content delivery distribution which takes about 35-40 minutes to fully set up, then messing around with a template that you have to wait where you can only find out if it's correct or not after 40 minutes to check is a great way to lose a whole day just fixing one tiny issue.

CFN Lint

Instead of waiting for CloudFormation to explode on invalid templates, use CFN Lint to perform basic validation locally. CFN Lint understands CloudFormation resources, so it is smart enough to catch when you mistype a property name. Install CFN Lint locally using the following command:

```
pip install cfn-lint
```

You can then check the basic syntax of your templates from the command line easily:

```
$ cfn-lint template.yaml
E0001 Error transforming template: Resource with id [HelloWorldFunction] i

s invalid. Missing required property 'Handler'.
template.yaml:1:1
```

`template.yaml` has been modified so you can see how cfn-lint detects errors. Press the **Run** button to see the output.

Environment Variables

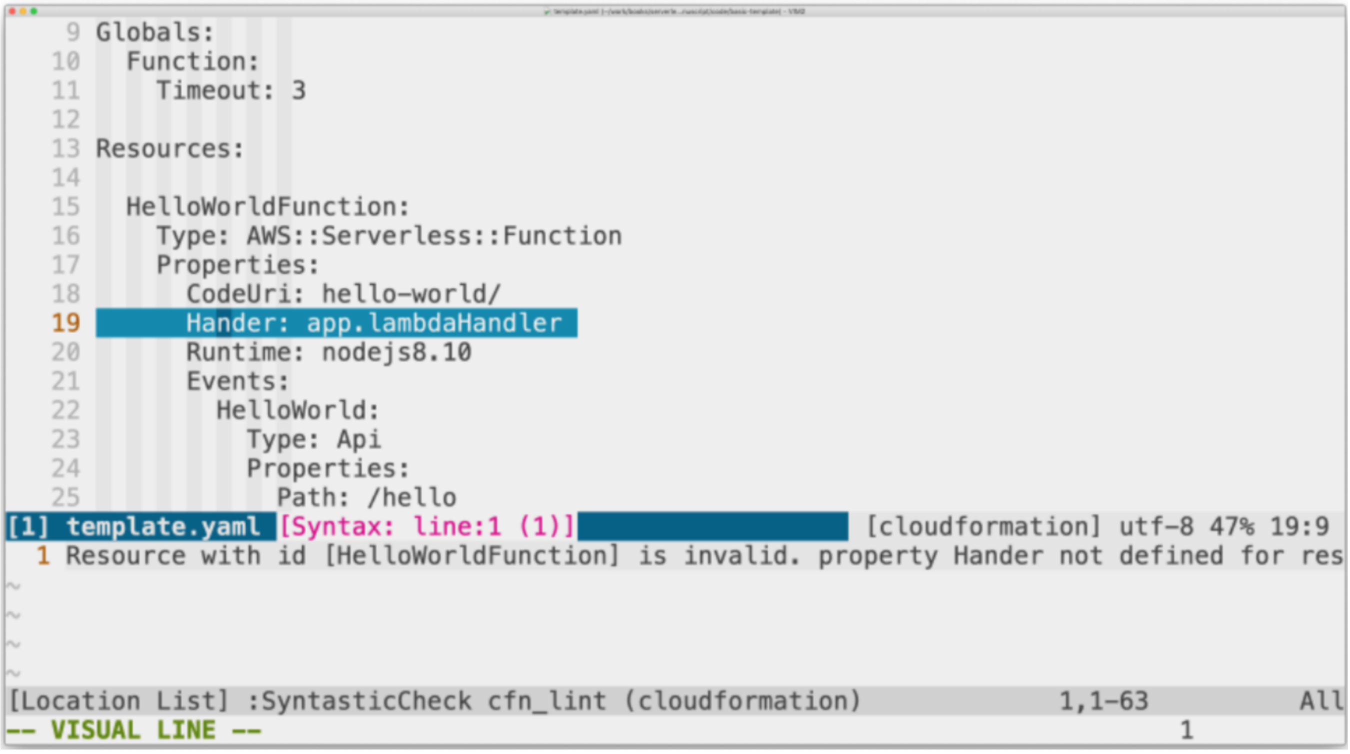
Key:	Value:
LANG	C.UTF-8
LC_ALL	C.UTF-8
AWS_ACCESS_KEY_ID	Not Specified...
AWS_SECRET_ACCE...	Not Specified...
BUCKET_NAME	Not Specified...
AWS_REGION	Not Specified...

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >
  sam-app

  Sample SAM Template for sam-app
Globals:
  Function:
    Timeout: 3 Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello-world/
      Handler: app.lambdaHandler
      Runtime: nodejs12.x
      Events:
        HelloWorld:
          Type: Api
          Properties:
            Path: /hello
            Method: get
Outputs:
  HelloWorldApi:
    Description: "API Gateway endpoint URL for Prod stage for Hello World function"
    Value: !Sub "https://${ServerlessRestApi}.execute-api.${AWS::Region}.amazonaws.com/Prod/h
  HelloWorldFunction:
    Description: "Hello World Lambda Function ARN"
    Value: !GetAtt HelloWorldFunction.Arn
  HelloWorldFunctionIamRole:
    Description: "Implicit IAM Role created for Hello World function"
    Value: !GetAtt HelloWorldFunctionRole.Arn
```

You can also modify the file as you wish. Press the **Run** button after modifying

You can also modify the file as you wish. Press the **Run** button after modifying the file. Afterwards, run `cfn-lint template.yaml` to validate your file.



```
Globals:
  Function:
    Timeout: 3

Resources:

  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello-world/
      Hander: app.lambdaHandler
      Runtime: nodejs8.10
      Events:
        HelloWorld:
          Type: Api
          Properties:
            Path: /hello
[1] template.yaml [Syntax: line:1 (1)]                  [cloudformation] utf-8 47% 19:9
  1 Resource with id [HelloWorldFunction] is invalid. property Hander not defined for res
~
~
~
~
[Location List] :SyntasticCheck cfn_lint (cloudformation)                1,1-63         All
-- VISUAL LINE --                                                         1
```

CFN lint saves us from simple typos.

Yamllint and ESlint

AWS resources have many optional configuration properties, and CFN Lint won't complain if YAML formatting issues turn an optional parameter into a non-existent value (wrong indentation is a common culprit). To complement CFN Lint, use yamllint for YAML or eslint for JSON to check for such generic formatting mistakes.

> ### SAM validation
>
> The SAM command-line tool also has a validation utility, `sam validate`. CFN Lint executes SAM validation internally, so it checks for both CloudFormation and SAM errors.

In the next lesson, you will learn how to work in a team while using AWS Lambda.