

DATE : 12.02.2025
DT/NT : DT
LESSON : AWS
SUBJECT: CLOUDFORMATION

BATCH : B 303

AWS-DEVOPS



TECHPRO
EDUCATION



techproeducation.com



+1 (585) 304 29 59





Table of Contents

- ▶ Introduction to AWS CLOUDFORMATION
- ▶ Templates
- ▶ Stacks
- ▶ YAML & JSON



Introduction to AWS CLOUDFORMATION

Introduction to AWS CLOUDFORMATION

What is CloudFormation?

- Create-Delete-Modify one or few AWS resources
 - Easy ✓
- Hundreds ?



Introduction to AWS CLOUDFORMATION

What is CloudFormation?



CloudFormation is an AWS service which enables you to create, manage, configure, replicate and delete AWS resources easily and rapidly using templates.

Introduction to AWS CLOUDFORMATION

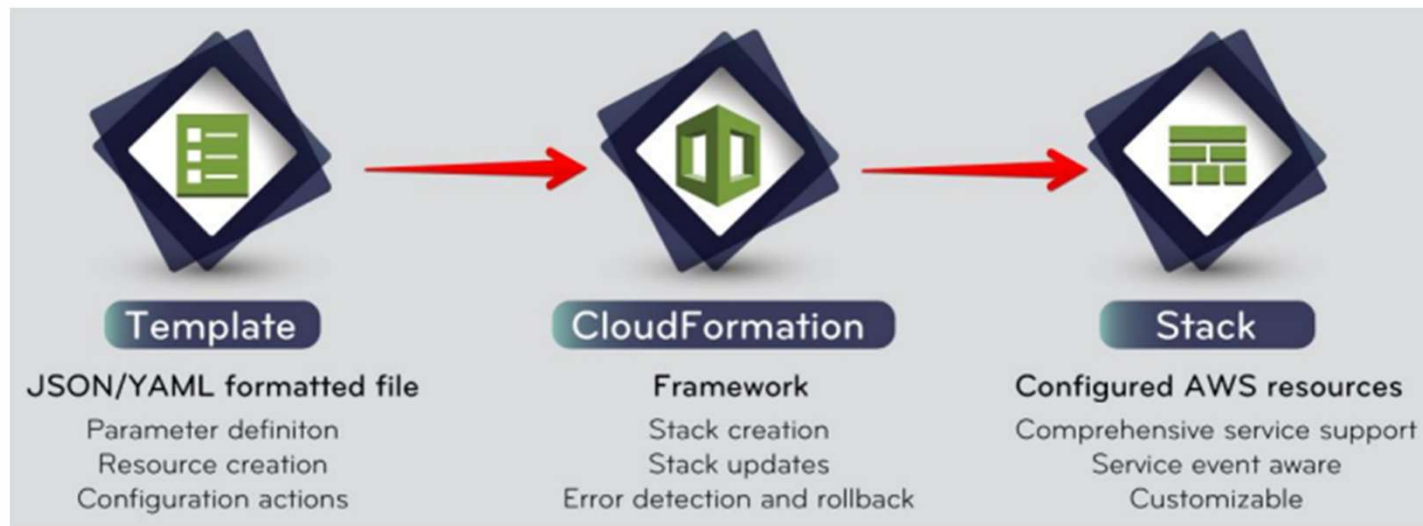
What is CloudFormation?



Provisions and manages stacks of AWS resources based on templates you create to model your infrastructure architecture.

Introduction to AWS CLOUDFORMATION

Templates and Stacks



- Templates and Stacks are the main components of AWS CloudFormation.

Introduction to AWS CLOUDFORMATION

Pricing



- You don't pay for using AWS CloudFormation.
- You pay for the resources you use.



Templates

Templates

What is a Template?

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "A simple EC2 instance",
  "Resources" : {
    "MyEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "ami-0ff8a91507f77f867",
        "InstanceType" : "t1.micro"
      }
    }
  }
}
```

- A template is a JSON or YAML formatted text file which you specify
- the AWS resources you want to create.



Templates

What is a Template?

- You create resources, define the parameters and configure your settings with templates.
- Templates can have the extensions .json, .yaml, .yml, .template, or .txt.

Templates

Structure

1. Template Version
2. Description
3. Metadata
4. Parameters
5. Mappings
6. Conditions
7. Transform
8. Resources
9. Outputs

```
{
  "AWSTemplateFormatVersion" : "version date",
  "Description" : "JSON string",
  "Metadata" : {
    template metadata
  },
  "Parameters" : {
    set of parameters
  },
  "Mappings" : {
    set of mappings
  },
  "Conditions" : {
    set of conditions
  },
  "Transform" : {
    set of transforms
  },
  "Resources" : {
    set of resources
  },
  "Outputs" : {
    set of outputs
  }
}
```

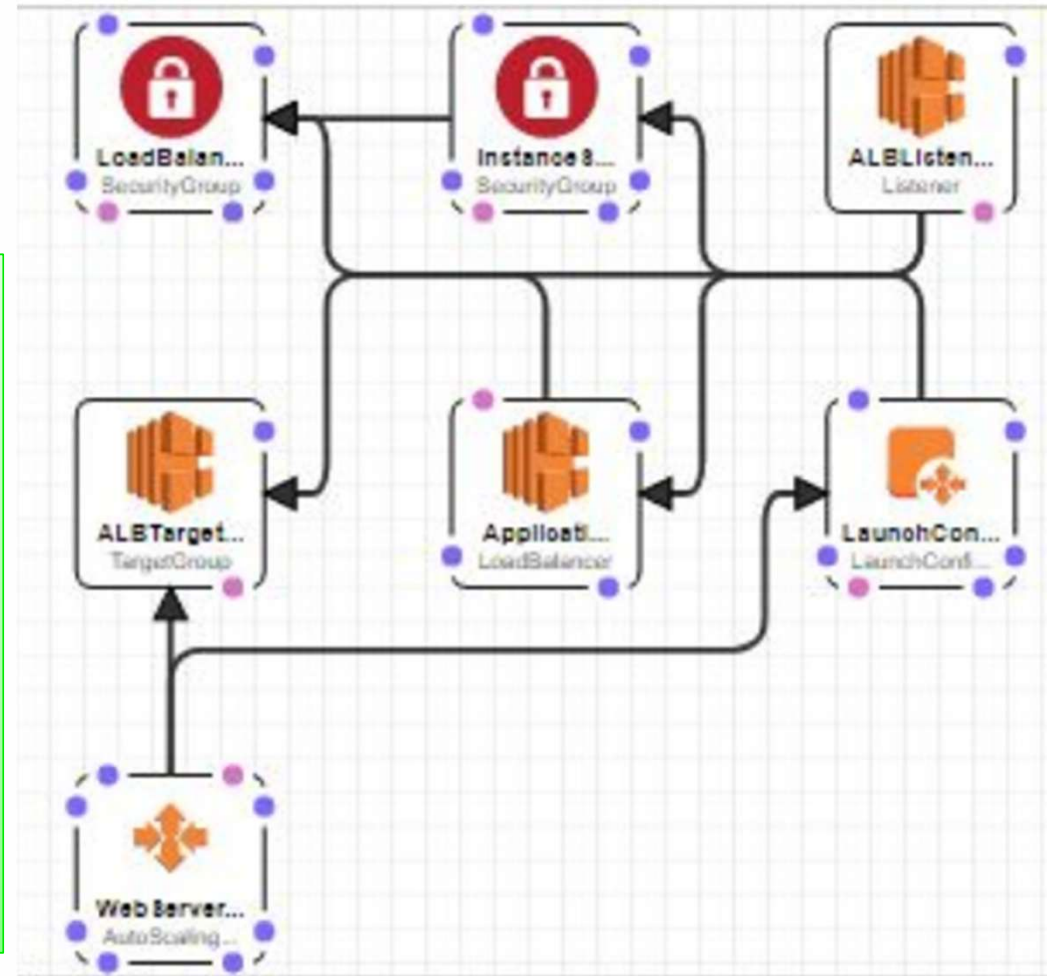
A decorative graphic to the left of the title, consisting of a large green right-angled triangle and a smaller blue right-angled triangle partially overlapping its left side.

Stacks

Stacks

What is a Stack?

- A stack is a single unit composed of the AWS resources provisioned by CloudFormation.
- All of the resources in a stack are defined and configured by the related template.



CloudFormation's Working Process

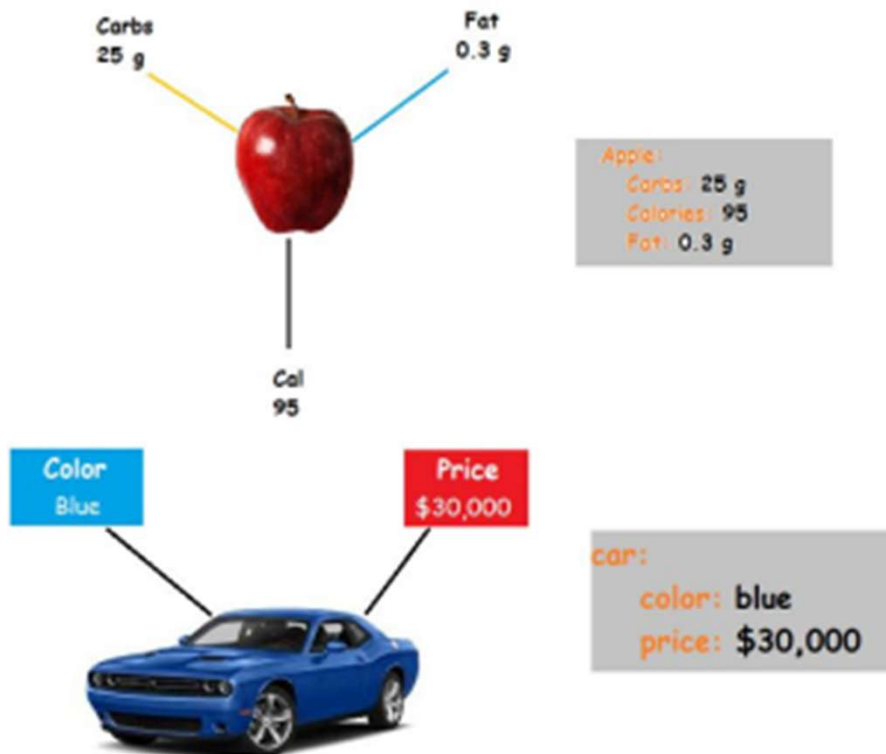




YAML & JSON

YAML & JSON

What is YAML?



YAML

```
1 yaml:
2   - slim and flexible
3   - better for configuration
4 object:
5   key: value
6   array:
7     - null_value:
8     - boolean: true
9     - integer: 1
10 paragraph: >
11   Blank lines denote
12
13   paragraph breaks
14 content: |-
15   Or we
16   can auto
17   convert line breaks
18   to save space
```

YAML & JSON

What is YAML?

- YAML (a recursive acronym for "YAML Ain't Markup Language") is a human-readable data-serialization standard for programming languages.
- It is commonly used for configuration files and in applications where data is being stored or transmitted.



YAML & JSON

What is YAML?

- The files should have .yaml (or .yml) as the extension.
- YAML is designed by Clark Evans, Ingy döt Net, and Oren Ben-Kiki. (2001).
- <https://yaml.org>



YAML & JSON

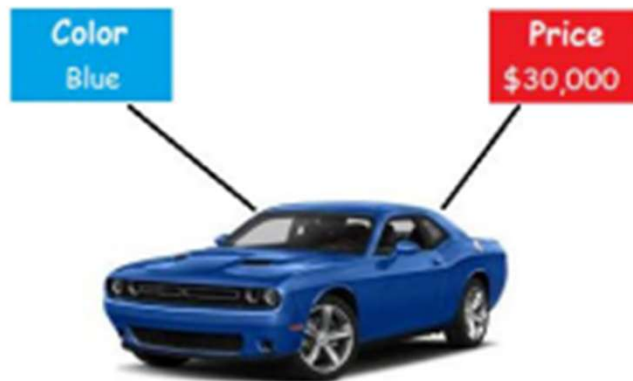
What is YAML?

- The key-value is YAML's basic building block.
- The key is always a string.
- The value is a scalar so that it can be various data types.



YAML & JSON

What is JSON?



```
{  
  "car": {  
    "color": "blue",  
    "price": "$30,000"  
  }  
}
```

JSON

```
1 {  
2   "json": [  
3     "rigid",  
4     "better for data interchange"  
5   ],  
6   "yaml": [  
7     "slim and flexible",  
8     "better for configuration"  
9   ],  
10  "object": {  
11    "key": "value",  
12    "array": [  
13      {  
14        "null_value": null  
15      },  
16      {  
17        "boolean": true  
18      },  
19      {  
20        "integer": 1  
21      }  
22    ]  
23  }  
24 }|
```

YAML & JSON

What is JSON?

- JSON stands for JavaScript Object Notation.
- JSON, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application; and also used for storing data.



YAML & JSON

What is JSON?

- The files should have .json as the extension.
- JSON is designed by Douglas Crockford (2001).
- <https://json.org/>



YAML & JSON

Quick Comparison

- YAML is best suited for configuration while JSON is better as a serialization format or serving up data for your APIs.
- YAML has a couple of big advantages including the ability to self-reference, support for complex datatypes, embedded block literals, comments, and more.
- Easily convertible to each other.

YAML & JSON

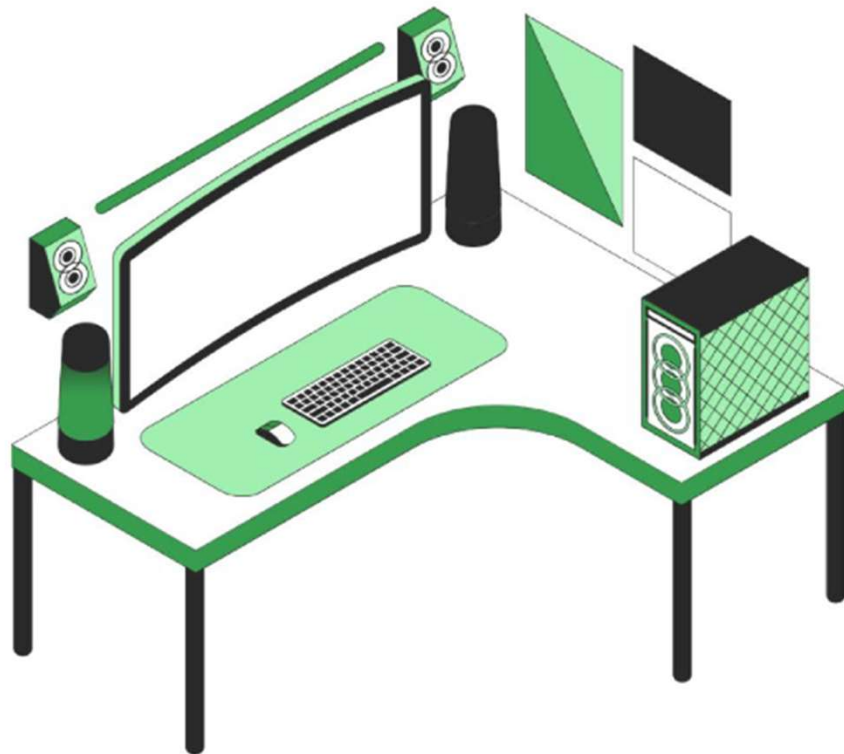
Quick Comparison

Type	YAML	JSON
Comments	Denoted with a hash/number sign	Not allowed
Hierarchy	Mappings, and sequences can be nested. Hierarchy is determined by the indentation level	Objects and arrays can be nested, and are denoted by braces and brackets, respectively.
Arrays	[first, second, 3]	["first", "second", 3]
Strings	Does not require quoting but supports both single and double quotes	Must be double-quoted. Allows character (tabs, newlines, etc.) escaping with a backslash as the escape character.
Numbers	Built-in support for integers, floating-point, octal and hexadecimal numbers	Floating point numbers in scientific notation. Infinity is not permitted.
Date/Timestamp	Supported	Not supported

YAML & JSON

Quick Comparison

JSON	YAML
<pre>1 { 2 "json": [3 "rigid", 4 "better for data interchange" 5], 6 "yaml": [7 "slim and flexible", 8 "better for configuration" 9], 10 "object": { 11 "key": "value", 12 "array": [13 { 14 "null_value": null 15 }, 16 { 17 "boolean": true 18 }, 19 { 20 "integer": 1 21 } 22] 23 } 24 }</pre>	<pre>1 --- 2 json: 3 - rigid 4 - better for data interchange 5 yaml: 6 - slim and flexible 7 - better for configuration 8 object: 9 key: value 10 array: 11 - null_value: 12 - boolean: true 13 - integer: 1 14</pre>



Do you have any questions?

Send it to us! We hope you learned something new.