

Template Anatomy

Resources	Mappings
Description	Conditions
Metadata	Transform
Parameters	Outputs

A Resources section is required.

Intrinsic Functions Syntax

YAML Shorthand	<code>! Ref arg</code>
YAML	<code>Fn::Ref arg</code>
JSON	<code>{ " Fn: :Re f": arg }</code>

Common Intrinsic Functions

`!FindInMap [Map, TopLevelKey, SecondLevelKey]`

`!GetAtt a.Arn`

`!ImportValue a`

`!Join [':', ['a', 'b']]`

`!Ref a`

`!Select ['1', ['a', 'b']]`

`!Split [':', 'a:b']`

`!Sub 'a-${b}'`

You can't nest the shorthand YAML functions. You must do:

`Fn::ImportValue: !Sub " ${a }-b "`

not `!ImportValue !Sub " ${a }-b "`

Transforms

Common Pseudo Parameters

<code>AWS::AccountId</code>	12-digit AWS account
<code>AWS::NoValue</code>	Use in conditionals
<code>AWS::Region</code>	Deployment region
<code>AWS::StackId</code>	ARN of the current stack
<code>AWS::StackName</code>	Name of the current stack

Reference pseudo parameters just like regular parameters, e.g. `!Ref AWS::Region`

Condition Functions

<code>!Equals ['a', 'b']</code>	false
<code>!And [!Equals ['a', 'a'], !Equals ['a', 'b']]</code>	false
<code>!Or [!Equals ['a', 'a'], !Equals ['a', 'b']]</code>	true
<code>!Not [!Equals ['a', 'a']]</code>	false
<code>!If [condition, 100, 10]</code>	100 if condition is true, 10 otherwise

Export Outputs

```

Outputs:
  MyVPCOutput:
    Value: !Ref VPCResource
    Export:
      Name: !Sub " ${AWS::StackName}-VPC_ID"
  
```

Produces a resource in this stack

- For each AWS account, Export names must be unique within a region.

- You can't create cross-stack references across regions.
- You can't delete a stack if another stack references one of its parameter or outputs.

reference export a from another stack

Produces 'b'

Produces ['a', 'b']

Inject the value of b into a string

```
'Fn::Transform':  
  - Name: 'AWS::Include'  
    Parameters:  
      Location: s3://bucket/snippet.yml
```

Use an Include transform to reference a template snippet stored separately from the main CloudFormation template.

Custom Resources

Use the `AWS::CloudFormation::CustomResource` or `Custom::String` resource type to define custom resources.

To create a custom resource, you need:

- **A template that includes a custom resource type.**
- **A custom resource provider** with a service token that the template developer uses.

During a stack operation, CloudFormation sends a request to a service token specified in the template, then waits for a response before proceeding.



By **forrestbrazeal**

cheatography.com/forrestbrazeal/

Not published yet.

Last updated 8th June, 2018.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>