Diogo Akio Balboni Miyake

84 Followers · About Follow

Introdução básica ao YAML para ansiosos ...

Diogo Akio Balboni Miyake Jun 17, 2019 · 4 min read

Last Update: 2020–07–04 (added example Docker compose)

...

Neste artigo abordarei de forma rápida (tentarei ir direto ao ponto) uma explicação básica da sintaxe YAML.

Quem nunca teve uma ansiedade de leve né...

Mas o que é YAML? é de comer?
Segundo o yaml.org:

> *What It Is: YAML is a human friendly data serialization standard for all programming languages.*

Ou seja: Um padrão de serialização de dados amigável para qualquer linguagem de programação.

“ O YAML foi criado especificamente para funcionar bem em casos de uso comum, como: arquivos de configuração, arquivos de log, mensagens entre processos, compartilhamento de dados entre linguagens, persistência de objetos e depuração de estruturas de dados complexas. Quando os dados são fáceis de visualizar e entender, a programação se torna uma tarefa mais simples.” (Documentação)

Resumidamente: É usado para arquivos de configuração assim como o JSON e o XML

Um exemplo prático são os templates do AWS CloudFormation e Cloudfront que usam este formato ou JSON, dê uma olhada aqui.

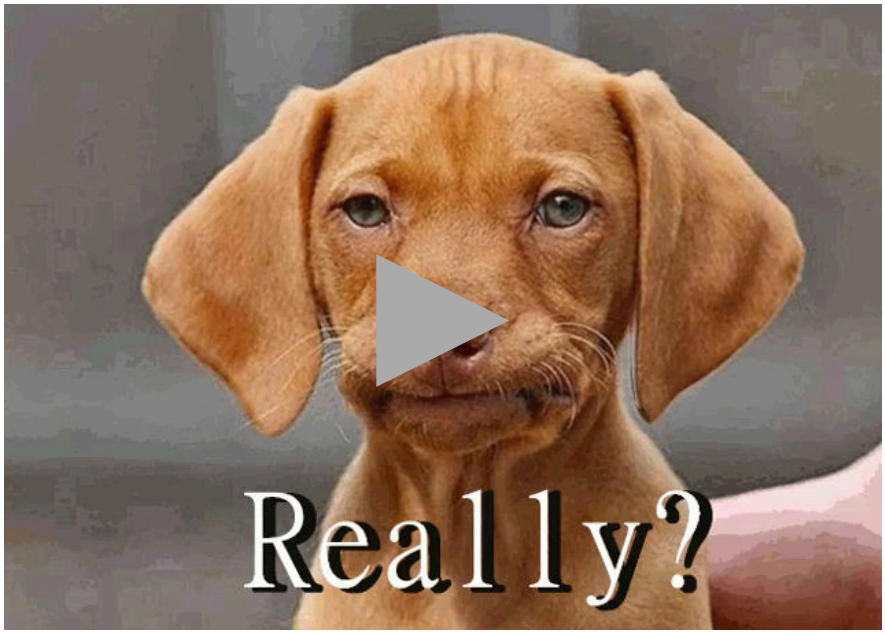
Só para exemplificar a facilidade do YAML comparado com o JSON para leitura humana:

```
# YAML

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://s3.amazonaws.com/cloudformation-
templates-us-east-1/S3_Bucket.template
      TimeoutInMinutes: '60'
Outputs:
  StackRef:
    Value: !Ref myStack
  OutputFromNestedStack:
    Value: !GetAtt myStack.Outputs.BucketName

# JSON
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myStack" : {
                "Type" : "AWS::CloudFormation::Stack",
                "Properties" : {
                   "TemplateURL" :
"https://s3.amazonaws.com/cloudformation-templates-us-east-
1/S3_Bucket.template",
               "TimeoutInMinutes" : "60"
                }
        }
    },
    "Outputs": {
       "StackRef": {"Value": { "Ref" : "myStack"}},
       "OutputFromNestedStack" : {
```

```
                "Value" : { "Fn::GetAtt" : [ "myStack",
"Outputs.BucketName" ] }
            }
        }
}
```



Sério... uma piadinha sem graça dessa...

Bora lá agora ver com funciona tudo isso

YAML tem a extensão … ".yaml"(jura? não brinca) e é case sensitive.

O primeiro caractere é o hash (#) use ele para comentar o arquivo.

```
# Para comentários basta usar o caractere (#).
# Yaml não permite
# multiplas linhas em comentário.
```

Pode se ter vários documentos com fluxos únicos, basta usar o (---) para iniciar o documento e (…) para finalizar.
Podemos ter **strings, null, integers, floats e booleanos.**
As strings podem ser com ou sem aspas duplas (“ ”) ou únicas (‘ ’), geralmente use elas quando se tem caracteres especiais

```
#Doc 1
--- # Start
tipo: "Teste" # ou 'Teste' ou teste
teste: true
numero: 1
nulo: null
... # End
```

Para data é seguido o padrão ISO 8601. (yyyy-mm- ddThh:mm:ss.ffffff)
Note que é importante indentar o código com espaços.

```
#Doc 2
---
pessoa:
 nome: &nome Fulano
 sobrenome: Cicrano
 idade: 45
 data_nascimento: 1974-05-13 09:25:55
 "estado civil": null
 masculino: true

 hobbies:
  - 'andar de bicicleta'
  - patinar
  - nadar
```

As listas podem conter *colchetes* ([]), e caso precisar pode usar uma estrutura parecida com dicionário em python ({}) com *chaves* para o item *‘chave:valor’* os membros da lista são indicados por (-) *hífen.*

```
 nadar: ["Praia", "Piscina", "Lago"]
amigos:
 - nome: "João"
   idade: 25
 - {nome: "Eduardo", idade: 24}                    -
```

```
  -
    nome: "Frederico"
    idade: 26
```

Caso queira que o YAML leia mais de uma linha em uma chave pode ser usado o caractere *maior* (>), no retorno ele junta tudo e retorna uma linha somente. Quando se usa o *pipe* (|) ele retorna com varias linhas de acordo com a estrutura que foi criada. Aqui tem uns exemplos bons.

```
descricao: >
  Aqui tem algumas caracteristicas do fulano,
  ele possui 45 anos e tem alguns amigos.
  Ele adora esportes.

 assinatura: |
  Fulano
  Cicrano Org
  email - cicrano.fulano@gmail.com
id: *nome
...
```

Ah e eu já ia esquecendo quando você tem nós (nesta sintaxe são itens que se repetirão) você pode usar o *caractere "e" comercial (&), no exemplo &dado* e para puxar a referência o caractere "asterisco"(*) neste caso fica **dado*, note no inicio acima o ***&nome*** e abaixo a ****nome*** na id.

> *Para maiores referências dos nós: clica aqui vai é rápido…*

Bom como eu disse é uma referência básica e rápida, abaixo tem o gist que criei e algumas referências bem legais que usei para poder estudar.

Para validar se seu YAML segue o padrão pode ser usado este site.

Como exemplo prático podemos ver esse docker-compose.yaml

Referências:

- https://yaml.org/
- https://www.tutorialspoint.com/yaml/yaml_basics.htm

- https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html
- https://rollout.io/blog/yaml-tutorial-everything-you-need-get-started/

Espero que tenham gostado, críticas para melhora são bem vindas.

Thanks !!

Yaml Programming Json AWS Syntax