



Administración de infraestructura en AWS (CloudFormation)

</> Índice

- ¿Qué es AWS CloudFormation?
- Ventajas de usar AWS CloudFormation
- Templates de AWS CloudFormation
- Lab - AWS CloudFormation

</> ¿Qué es AWS CloudFormation?

- AWS CloudFormation es un servicio que nos permite gestionar, configurar y provisionar nuestra infraestructura en AWS como código
- Los recursos se definen utilizando templates de CloudFormation
- CloudFormation interpreta los templates y hace llamadas a las APIs de AWS para crear los recursos definidos
- Acepta formatos YAML y JSON



</> Ventajas de usar AWS CloudFormation

- El provisionamiento de infraestructura es consistente y menos propenso a errores
- Es mucho más rápido y requiere menos esfuerzo que configurar todo manualmente
- Podemos versionar el código que define la infraestructura
- Es gratis, sólo tiene coste la infraestructura creada
- Podemos utilizarlo para gestionar actualizaciones y dependencias
- Podemos hacer rollback en caso de errores en despliegues

</> Templates de AWS CloudFormation

```
{
  "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
  }
}
```

```
---
AWSTemplateFormatVersion: "version date"

Description:
  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
```

</> Templates de AWS CloudFormation

- AWSTemplateFormatVersion

JSON

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

YAML

```
AWSTemplateFormatVersion: "2010-09-09"
```

</> Templates de AWS CloudFormation

- Description

JSON

```
"Description" : "Here are some details about the template."
```

YAML

```
Description: >  
  Here are some  
  details about  
  the template.
```

</> Templates de AWS CloudFormation

- Metadata

JSON

```
"Metadata" : {  
  "Instances" : {"Description" : "Information about the instances"},  
  "Databases" : {"Description" : "Information about the databases"}  
}
```

YAML

```
Metadata:  
  Instances:  
    Description: "Information about the instances"  
  Databases:  
    Description: "Information about the databases"
```


</> Templates de AWS CloudFormation

- Parameters

JSON

```
"Parameters" : {  
  "InstanceTypeParameter" : {  
    "Type" : "String",  
    "Default" : "t2.micro",  
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"],  
    "Description" : "Enter t2.micro, m1.small, or m1.large. Default is t2.micro."  
  }  
}
```

YAML

```
Parameters:  
  InstanceTypeParameter:  
    Type: String  
    Default: t2.micro  
    AllowedValues:  
      - t2.micro  
      - m1.small  
      - m1.large  
    Description: Enter t2.micro, m1.small, or m1.large. Default is t2.micro.
```

</> Templates de AWS CloudFormation

- Conditions

JSON

```
"Conditions" : {  
  "Logical ID" : {Intrinsic function}  
}
```

YAML

```
Conditions:  
  Logical ID:  
    Intrinsic function
```

</> Templates de AWS CloudFormation

- Mappings

JSON

```
"Mappings" : {  
  "Mapping01" : {  
    "Key01" : {  
      "Name" : "Value01"  
    },  
    "Key02" : {  
      "Name" : "Value02"  
    },  
    "Key03" : {  
      "Name" : "Value03"  
    }  
  }  
}
```

YAML

```
Mappings:  
  Mapping01:  
    Key01:  
      Name: Value01  
    Key02:  
      Name: Value02  
    Key03:  
      Name: Value03
```

</> Templates de AWS CloudFormation

- Transform

JSON

```
{
  "Transform" : {
    "Name" : "AWS::Include",
    "Parameters" : {
      "Location" : "s3://MyAmazonS3BucketName/MyFileName.json"
    }
  }
}
```

YAML

```
Transform:
  Name: 'AWS::Include'
  Parameters:
    Location: 's3://MyAmazonS3BucketName/MyFileName.yaml'
```

</> Templates de AWS CloudFormation

- Resources

JSON

```
"Resources" : {  
  "Logical ID" : {  
    "Type" : "Resource type",  
    "Properties" : {  
      Set of properties  
    }  
  }  
}
```

YAML

```
Resources:  
  Logical ID:  
    Type: Resource type  
    Properties:  
      Set of properties
```

</> Templates de AWS CloudFormation

- Outputs

JSON

Use braces to enclose all output declarations. Delimit multiple outputs with commas.

```
"Outputs" : {  
  "Logical ID" : {  
    "Description" : "Information about the value",  
    "Value" : "Value to return",  
    "Export" : {  
      "Name" : "Value to export"  
    }  
  }  
}
```

YAML

```
Outputs:  
  Logical ID:  
    Description: Information about the value  
    Value: Value to return  
    Export:  
      Name: Value to export
```

</> Templates de AWS CloudFormation

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  License: Apache-2.0
Description: 'Create an Amazon EC2 instance running the Amazon Linux AMI'
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t3.small
    AllowedValues: [t2.nano, t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge]
    ConstraintDescription: must be a valid EC2 instance type.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  LatestAmiId:
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref 'InstanceType'
      SecurityGroups: [!Ref 'InstanceSecurityGroup']
      KeyName: !Ref 'KeyName'
      ImageId: !Ref 'LatestAmiId'
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access via port 22
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref 'SSHLocation'
Outputs:
  InstanceId:
    Description: InstanceId of the newly created EC2 instance
    Value: !Ref 'EC2Instance'
```



¿Cuál es la mejor manera de aprender sobre AWS Cloudwatch?

"For the things we have to learn before we can do them, we learn by doing them"

Aristóteles

</> Lab - AWS CloudFormation

¿Qué hemos aprendido?

- Cómo crear un Stack de AWS CloudFormation desde un template generado por nosotros
- Cómo actualizar un Stack de AWS CloudFormation creado para cambiar la infraestructura
- Cómo borrar un Stack de AWS CloudFormation

</> Consejos - CloudFormation

- Validar los templates antes de desplegarlos
- Podemos implementar nuestros recursos propios
- Habilitar la protección en la destrucción de recursos
- Aplicar clean code a nuestros templates
- Utilizar yaml mejor que json

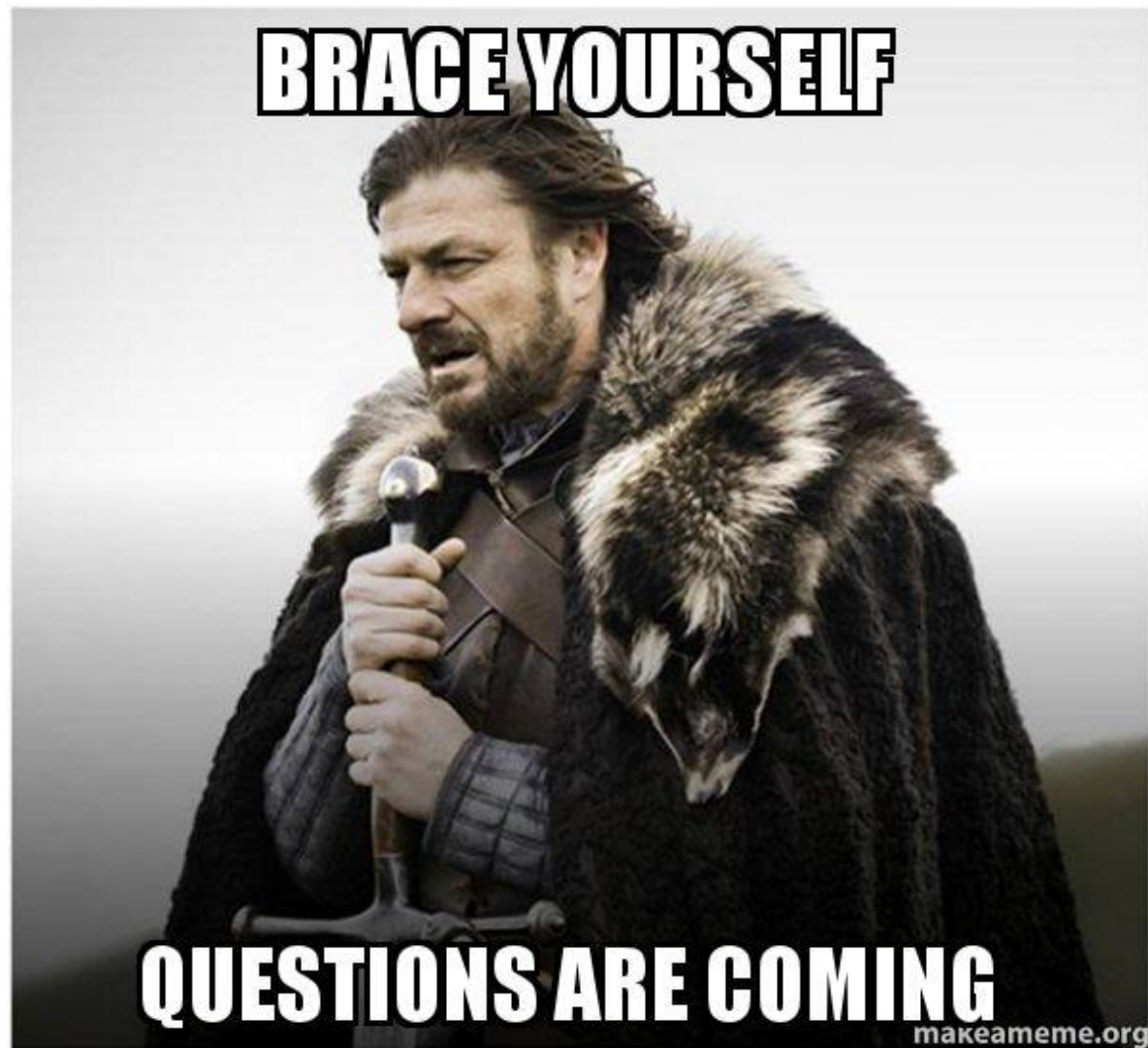
</> Importante - CloudFormation

- La gestión de errores en los despliegues no es muy buena
- Es complejo de utilizar para arquitecturas a gran escala
- Hay buenas alternativas como Terraform
- Algunas operaciones son muy lentas
- Podemos sobrepasar los límites de las APIs de AWS

¿Qué hemos aprendido en este módulo?

- Conocer qué es AWS CloudFormation
- Conocer las ventajas de utilizar AWS CloudFormation
- Entender cómo se crean los templates de AWS CloudFormation
- Aprender a usar el servicio de AWS CloudFormation de AWS


</> Preguntas





- Unai Arrien
- Email de contacto: *unai.arrien@gmail.com*

info@devacademy.es 

687374918 

@DevAcademyES 