

**4.3**

**Plantillas :  
AWS  
Cloudformation**

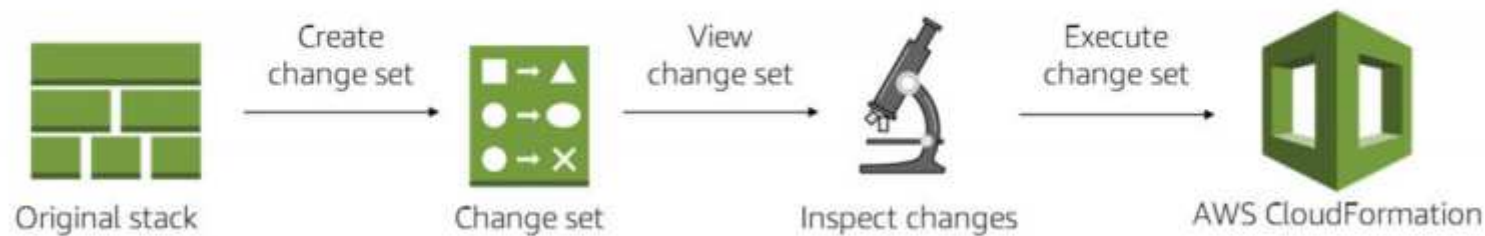
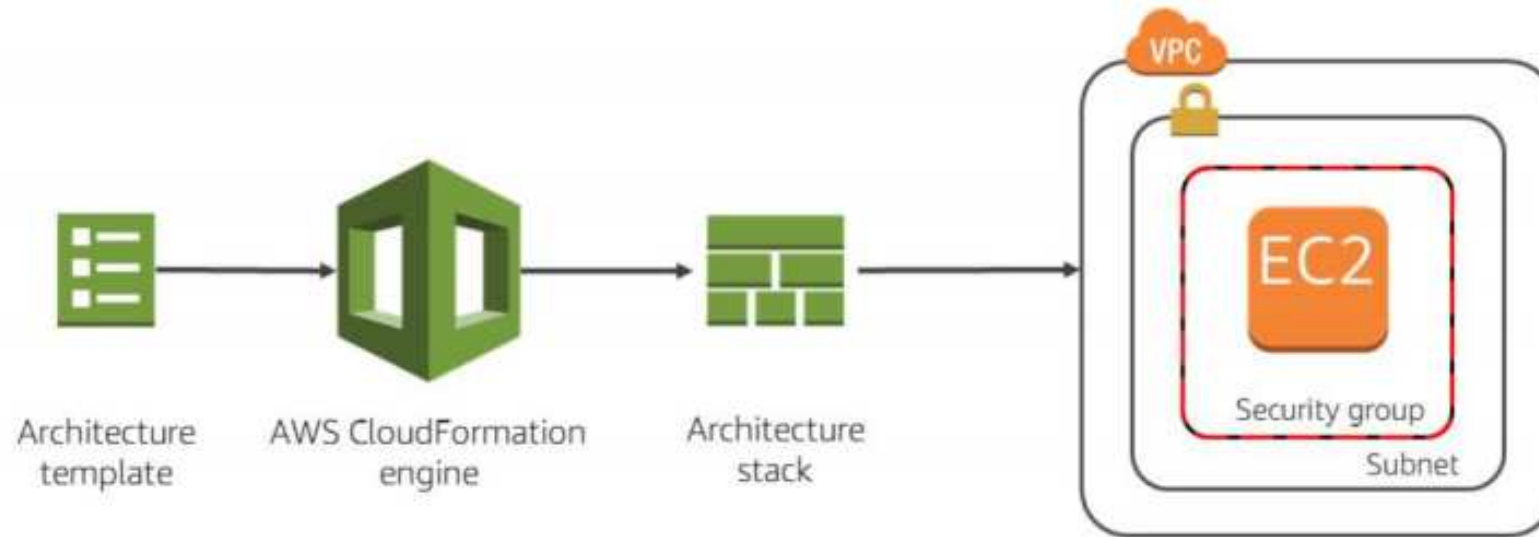
# **Plantillas : AWS Cloudformation**

---

- Plantillas : AWS Cloudformation
- Es un servicio de automatización de AWS
- Permite crear infraestructuras fácilmente y manera automatizada
- Se pueden crear cualquier tipo de recursos AWS
- Permite replicar fácilmente una infraestructura en otra región
- Infrastructure as Code : Yaml/Json
  - Permite realizar una trazabilidad y control de los cambios

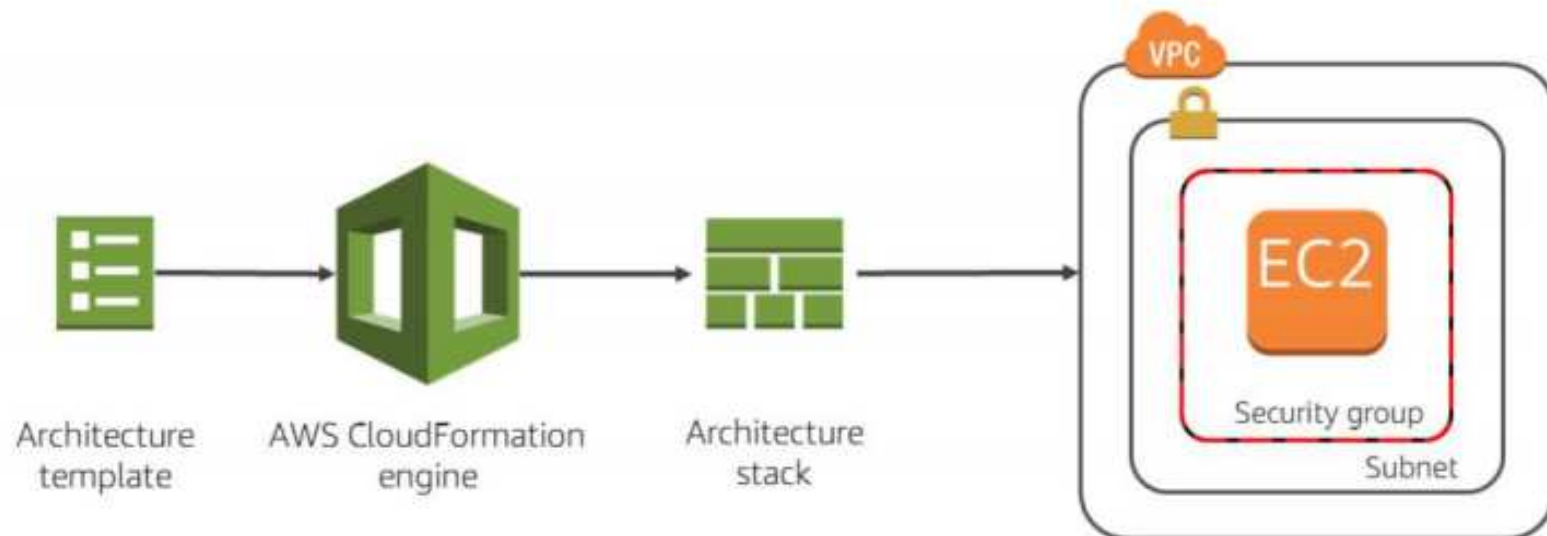
# Plantillas : AWS Cloudformation

- Plantillas : AWS Cloudformation



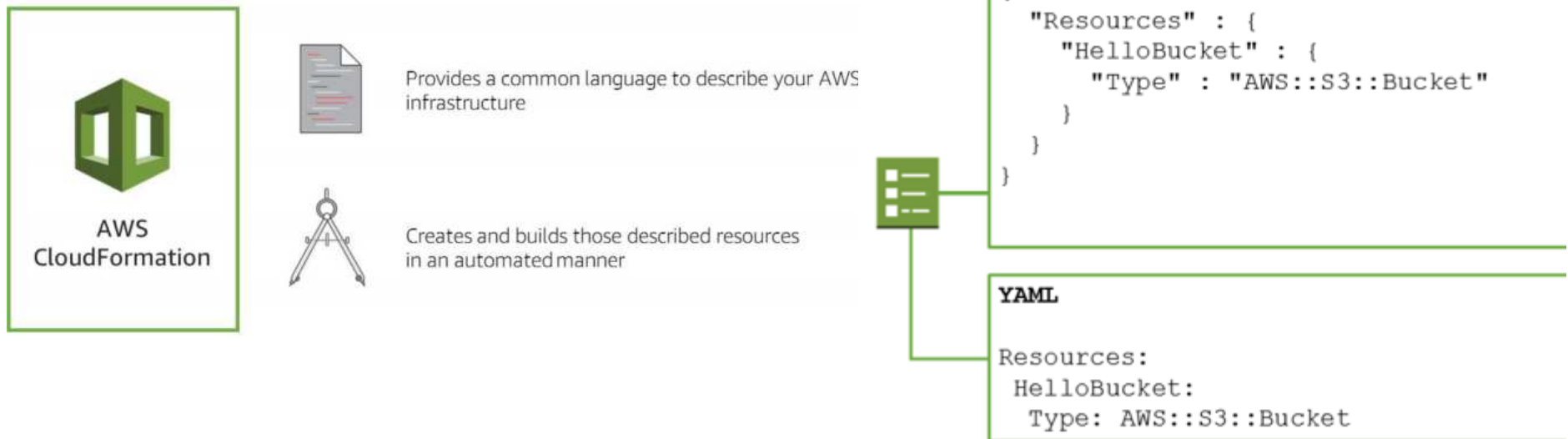
# Plantillas : AWS Cloudformation

- Plantillas : AWS Cloudformation



# Plantillas : AWS Cloudformation

- Plantillas : AWS Cloudformation
- Es un servicio ofrecido por AWS
- Permite definir toda la infraestructura en un fichero Json/Yaml
  - Único click/comando para desplegar/eliminar la infraestructura



# Plantillas : AWS Cloudformation

- Plantillas : AWS Cloudformation
- Crear un stack mediante CLI

```
aws cloudformation create-stack --stack-name vpn --template-url  
https://s3.amazonaws.com/bucketname/automation/vpn-cloudformation.json
```

- Esperar creación

```
while [[ `aws cloudformation describe-stacks --stack-name vpn --query  
Stacks[0].StackStatus` != *"COMPLETE"* ]]  
do  
    sleep 10  
done
```

- Visualizar datos de salida

```
aws cloudformation describe-stacks --stack-name vpn --query Stacks[0].Outputs
```

# Plantillas : AWS Cloudformation

- Plantillas : AWS Cloudformation
- Estructura de una plantilla

---

**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

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Recursos y referencias
- S3.yaml

**AWSTemplateFormatVersion:** 2010-09-09

**Description:** Basic S3 Bucket CloudFormation template

**Resources:**

**S3BucketForWebsiteContent:**

**Type:** AWS::S3::Bucket

**Properties:**

**AccessControl:** PublicRead

**Outputs:**

**BucketName:**

**Value:** *!Ref S3BucketForWebsiteContent*

**Description:** Name of the newly created Amazon S3 Distribution



# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Pasos para crear el stack
  - Crear un bucket para alojar los cloudformation

```
vagrant@ubuntu-xenial:~$ aws s3 mb s3://mgep-master-automation  
make_bucket: mgep-master-automation
```

- Subir la plantilla a s3

```
$ aws s3 cp s3simple.yaml s3://mgep-master-automation/cloudformation/  
upload: ./s3simple.yaml to s3://mgep-master-automation/cloudformation/s3simple.yaml
```

```
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws s3 ls --recursive s3://mgep-master-automation  
2019-10-09 10:07:20          335 cloudformation/s3simple.yaml
```

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Pasos para crear el stack
  - Agregar acceso publico al objeto

```
{  
  "Version":"2012-10-17",  
  "Statement":[  
    {  
      "Sid":"AddPerm",  
      "Effect":"Allow",  
      "Principal": "*",  
      "Action":["s3:GetObject"],  
      "Resource":["arn:aws:s3:::mgcp-master-  
automation/cloudformation/*"]  
    }  
  ]  
}
```

\$aws s3api put-bucket-policy --bucket mgcp-master-automation --policy file://bucketpolicy.json

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Pasos para crear el stack
  - Crear stack Cloudformation

```
aws cloudformation create-stack --stack-name s3simple --template-url https://mgep-master-automation.s3.amazonaws.com/cloudformation/s3simple.yaml
```

```
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws cloudformation create-stack --stack-name s3simple --template-url https://mgep-master-automation.s3.amazonaws.com/cloudformation/s3simple.yaml
{
  "StackId": "arn:aws:cloudformation:eu-west-1:420693608596:stack/s3simple/60dbf0-ea7d-11e9-b7aa-0a259d6932f8"
}
```

- Ver info del estado del stack

```
aws cloudformation describe-stacks --stack-name vpn --query Stacks[0].StackStatus
```

```
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws cloudformation describe-stacks --stack-name s3simple --query Stacks[0].StackStatus
"CREATE_COMPLETE"
```

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Ver output del stack

aws cloudformation describe-stacks --stack-name vpn --query  
Stacks[0].Outputs

```
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws cloudform
[
  {
    "Description": "Name of the newly created Amazon S3 Distribution",
    "OutputKey": "BucketName",
    "OutputValue": "s3simple-s3bucketforwebsitecontent-mpffvj76as9v"
  }
]
```

```
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws s3 ls
2019-10-02 10:13:23 cf-templates-6smw15w7vf9n-eu-west-1
2019-09-17 15:04:03 cf-templates-6smw15w7vf9n-us-east-1
2019-02-08 16:22:39 cf-templates-6smw15w7vf9n-us-west-2
2019-10-02 13:38:22 elasticbeanstalk-eu-west-1-420693608596
2019-09-12 12:32:40 facturas-jaagibas-master
2019-09-12 12:33:19 facturas-master-jaagibas
2019-10-02 10:27:34 master-imagery
2019-10-09 11:46:05 mgep-master-automation
2019-10-09 11:49:41 s3simple-s3bucketforwebsitecontent-mpffvj76as9v
2019-09-30 14:50:51 wordpress-master-practicas
```

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Eliminar Stack

`aws cloudformation delete-stack --stack-name s3simple`

```
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws s3 ls
2019-10-02 10:13:23 cf-templates-6smw15w7vf9n-eu-west-1
2019-09-17 15:04:03 cf-templates-6smw15w7vf9n-us-east-1
2019-02-08 16:22:39 cf-templates-6smw15w7vf9n-us-west-2
2019-10-02 13:38:22 elasticbeanstalk-eu-west-1-420693608596
2019-09-12 12:32:40 facturas-jaagibas-master
2019-09-12 12:33:19 facturas-master-jaagibas
2019-10-02 10:27:34 master-imagery
2019-10-09 11:46:05 mgep-master-automation
2019-09-30 14:50:51 wordpress-master-practicas
```

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Aquí un ejemplo de una política más restrictiva de bucket

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AddPerm",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::45454646:user/kudeatzailea"},
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::mgcp-master-automation/cloudformation/*"]
    }
  ]
}
```

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de un bucket S3
- Como evitar que un recurso se elimine al eliminar un stack
- DeletionPolicy: Retain

```
AWSTemplateFormatVersion: 2010-09-09
Description: Basic S3 Bucket CloudFormation template
Resources:
  S3BucketForWebsiteContent:
    Type: AWS::S3::Bucket
    DeletionPolicy: Retain
    Properties:
      AccessControl: PublicRead
Outputs:
  BucketName:
    Value: !Ref S3BucketForWebsiteContent
    Description: Name of the newly created Amazon S3 Distribution
```

# Plantillas : AWS Cloudformation

- Funciones típicas
  - !Ref
    - PMOWNIP: !Ref "PMOWNIP"
  - !FindInMap
    - PRegStorage: !FindInMap ["RegionMap", !Ref "AWS::Region", "AStorage"]
  - !GetAttr
    - !GetAtt "MyVPC.Outputs.PrivateSubnets"
  - !Sub
    - !Sub "\${PMTemplateURL}/webapp-rds.yaml"



# Plantillas : AWS Cloudformation

- Ejemplo : Creación de una instancia EC2
- **Cloudformation : Recursos , referencias, parámetros, funciones y mappings**
- Parámetros:
  - Tipo de instancia
  - Nombre de clave de acceso
- Tipos de recursos a crear:
  - AWS::EC2::Instance
  - AWS::EC2::SecurityGroup
- Mappings
  - Tipo de AMI por region
- Salidas

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de una instancia EC2

---

AWSTemplateFormatVersion: "2010-09-09"

Description: "Example EC2 Set-up"

## Parameters:

InstanceType:

Description: "Enter t2.micro or m1.small. Default is t2.micro."

Type: "String"

Default: "t2.micro"

AllowedValues:

- "t2.micro"

- "m1.small"

KeyName:

Description: "Enter an existing EC2 KeyPair. Default is MyEC2Key"

Type: "String"

Default: "MyEC2Key"

# Plantillas : AWS Cloudformation

## • Ejemplo : Creación de una instancia EC2

### Mappings:

#### *RegionMap:*

eu-west-1:

AMI: "ami-bff32ccc"

ap-southeast-1:

AMI: "ami-dc9339bf" # My-Default-AMI

ap-southeast-2:

AMI: "ami-48d38c2b"

eu-central-1:

AMI: "ami-bc5b48d0"

ap-northeast-1:

AMI: "ami-383c1956"

us-east-1:

AMI: "ami-0b33d91d"

sa-east-1:

AMI: "ami-6817af04"

us-west-1:

AMI: "ami-d5ea86b5"

us-west-2:

AMI: "ami-f0091d91"

```
Resources:
  Ec2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      KeyName:
        Ref: "KeyName"
      SecurityGroups:
        - Ref: "InstanceSecurityGroup"
      InstanceType:
        Ref: "InstanceType"
      # Select the correct AMI to load (based on the region the s
      ImageId:
        Fn::FindInMap:
          - "RegionMap"
          - Ref: "AWS::Region"
          - "AMI"
```

# Plantillas : AWS Cloudformation

## —• Ejemplo : Creación de una instancia EC2

### Resources:

#### Ec2Instance:

**Type:** "AWS::EC2::Instance"

#### Properties:

KeyName:

Ref: "KeyName"

SecurityGroups:

- Ref: "InstanceSecurityGroup"

InstanceType:

Ref: "InstanceType"

#### ImageId:

Fn::FindInMap:

- "RegionMap"

- Ref: "AWS::Region"

- "AMI"

#### UserData:

Fn::Base64: !Sub |

#!/bin/bash -xe

yum install httpd -y

yum update -y

service httpd start

chkconfig httpd on

echo "<html><h1>Successfully Created EC2 via CF</h1></html>" > /var/www/html/index.html

# Plantillas : AWS Cloudformation

## • Ejemplo : Creación de una instancia EC2

### **InstanceSecurityGroup:**

Type: "AWS::EC2::SecurityGroup"

Properties:

GroupDescription: "Enable Access to our SSH access via port 22"

SecurityGroupIngress:

- IpProtocol: "tcp"

FromPort: "22"

ToPort: "22"

CidrIp: "0.0.0.0/0"

### **Outputs:**

ServerIP:

#Value: Fn::GetAtt: [ logicalNameOfResource, attributeName ]

Value: !GetAtt: [ Ec2Instance, PublicDnsName ]

Description: Server Public IP

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de una instancia EC2
- Ejecutar el stack

```
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws s3 cp ./EC2firstExample.1.yaml s3://mgep-master-automation/cloudformation/
upload: ./EC2firstExample.1.yaml to s3://mgep-master-automation/cloudformation/EC2firstExample.1.yaml
vagrant@ubuntu-xenial:/vagrant/practica_scripting/cloudformation$ aws cloudformation create-stack --stack-name ec2froga --template-url https://mgep-master-automation.s3.amazonaws.com/cloudformation/EC2firstExample.1.yaml
{
  "StackId": "arn:aws:cloudformation:eu-west-1:420693608596:stack/ec2froga/7da61800-ea93-11e9-a0d7-068bfa33c7ce"
}
```

- Como ver lo que ha ocurrido si ha habido rollback

`aws cloudformation describe-stack-events --stack-name ec2froga`

# Plantillas : AWS Cloudformation

- Ejemplo : Creación de una instancia EC2
- Como ver lo que ha ocurrido si ha habido rollback

aws cloudformation describe-stack-events --stack-name ec2froga

```
"StackName": "ec2froga",  
"PhysicalResourceId": "arn:aws:cloudformation:eu-west-1:420693608596:stack/ec2froga/7da61800-ea93-11e9-a0",  
"LogicalResourceId": "ec2froga"  
},  
{  
  "StackId": "arn:aws:cloudformation:eu-west-1:420693608596:stack/ec2froga/7da61800-ea93-11e9-a0d7-068bfa33",  
  "EventId": "Ec2Instance-CREATE_FAILED-2019-10-09T12:55:46.163Z",  
  "ResourceStatus": "CREATE_FAILED",  
  "ResourceType": "AWS::EC2::Instance",  
  "Timestamp": "2019-10-09T12:55:46.163Z",  
  "ResourceStatusReason": "The key pair 'MyEC2Key' does not exist (Service: AmazonEC2; Status Code: 400; Er  
b-ec52-4ea5-958b-bfc3f8e55dfc)",  
  "StackName": "ec2froga",  
  "ResourceProperties": "{\n    \"KeyName\": \"MyEC2Key\",  
    \"SecurityGroups\": [\"ec2froga-InstanceSecurityGroup-17N  
5BpbnN0YWxsIGh0dHBkIC15Cn11bSB1cGRhdGUgLXkKc2VydmljZSBodHRwZCBzdGFydApjaGtjb25maWcgaHR0cGQgb24KZWNoYAiPGh0bWw+PGg  
0Y8L2gxPjwvaHRtbD4iID4gL3Zhci93d3cvaHRtbC9pbmRleC5odG1sCg==\",  
    \"ImageId\": \"ami-bff32ccc\",  
    \"InstanceType\": \"t2.m  
  \"PhysicalResourceId\": \"\",  
  \"LogicalResourceId\": \"Ec2Instance\"
```

- ```
#!/bin/bash
mykey=vpc=$(aws ec2 describe-key-pairs --query "KeyPairs[0].KeyName" --output text)
aws cloudformation create-stack --stack-name ec2froga \
--template-url https://mgep-master-automation.s3.amazonaws.com/cloudformation/EC2firstExample.1.yaml \
--parameters ParameterKey=KeyName,ParameterValue=$mykey

while [[ `aws cloudformation describe-stacks --stack-name ec2froga --query Stacks[0].StackStatus` != *"COMPLETE"* ]]
do
    sleep 10
done
aws cloudformation describe-stacks --stack-name vpn --query Stacks[0].Outputs
```

```
{
  "StackId": "arn:aws:cloudformation:eu-west-1:420693608596:stack/ec2froga/61faabd0-ea97-11e9-b59f-02078729aa3e"
}
{
  "Description": "Server Public IP",
  "OutputKey": "ServerIP",
  "OutputValue": "ec2-52-210-134-124.eu-west-1.compute.amazonaws.com"
}
```

[illegible]



# Plantillas : AWS Cloudformation

- Ejemplo : Creación de una instancia EC2
  - Por ssh nos conectamos perfectamente pero no hay acceso web por el security group.
  - Modificarlo y volver a desplegar actualizando el stack
  - Update-stack

```
aws cloudformation update-stack --stack-name ec2froga \  
  --template-url https://mgep-master-  
automation.s3.amazonaws.com/cloudformation/EC2firstEx  
ample.1.yaml \  
  --parameters  
ParameterKey=KeyName,ParameterValue=$mykey
```

```
InstanceSecurityGroup:  
  Type: "AWS::EC2::SecurityGroup"  
  Properties:  
    GroupDescription: "Enable Access to our SSH access via p  
    SecurityGroupIngress:  
      - IpProtocol: "tcp"  
        FromPort: "22"  
        ToPort: "22"  
        CidrIp: "0.0.0.0/0"  
      - IpProtocol: "tcp"  
        FromPort: "80"  
        ToPort: "80"  
        CidrIp: "0.0.0.0/0"
```

← → ↻ ⓘ No es seguro | ec2-52-210-134-124.eu-west-1.compute.amazonaws.com

**Successfully Created EC2 instance via CF**

```
ec2-user@ip-172-31-12-195:~  
Using username "ec2-user".  
Authenticating with public key  
_ _ | _ _ | _ _ |  
_ _ | _ _ | _ _ | Amazon Lin  
https://aws.amazon.com/amazon-1  
[ec2-user@ip-172-31-12-195 ~]$
```

# Plantillas : AWS Cloudformation

- Utilizar variables de entorno
  - Por ejemplo para diferenciar dos entornos de trabajo

```
Parameters:
- LinuxAmiId:
  - Type: 'AWS::SSM::Parameter::Value<A
  - Default: '/aws/service/ami-amazon-l
- ENV:
  - Type: String
  - Default: dev
  - AllowedValues:
    - dev
    - prod
```

```
Mappings:
- EC2TypeConfig:
  - prod:
    - InstanceType: t2.small
  - dev:
    - InstanceType: t2.micro
```

```
Properties:
- Tags:
  - Key: Name
  - Value: Web-server -- port 80 and 22
- ImageId: !Ref LinuxAmiId
- InstanceType: !FindInMap[EC2TypeConfig, !Ref ENV, InstanceType]
```

# Plantillas : AWS Cloudformation

- Otros recursos : Añadir LoadBalancer

```
ElasticLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Subnets:
      - !Ref VCPublicSubnetId
    SecurityGroups:
      - !Ref WebServerSecurityGroup
    Listeners:
      - LoadBalancerPort: 80
        InstancePort: 80
        Protocol: HTTP
    HealthCheck:
      Target: HTTP:80/
      HealthyThreshold: 3
      UnhealthyThreshold: 5
      Interval: 30
      Timeout: 5
```

# Plantillas : AWS Cloudformation

- Ejercicio : AWS Cloudformation
- qwiklabs

← Launching and Managing a Web Application with AWS CloudFormation



Start Lab

01:50:00

## Launching and Managing a Web Application with AWS CloudFormation

1 hour 50 minutes

15 Credits

★★★★☆ Rate Lab

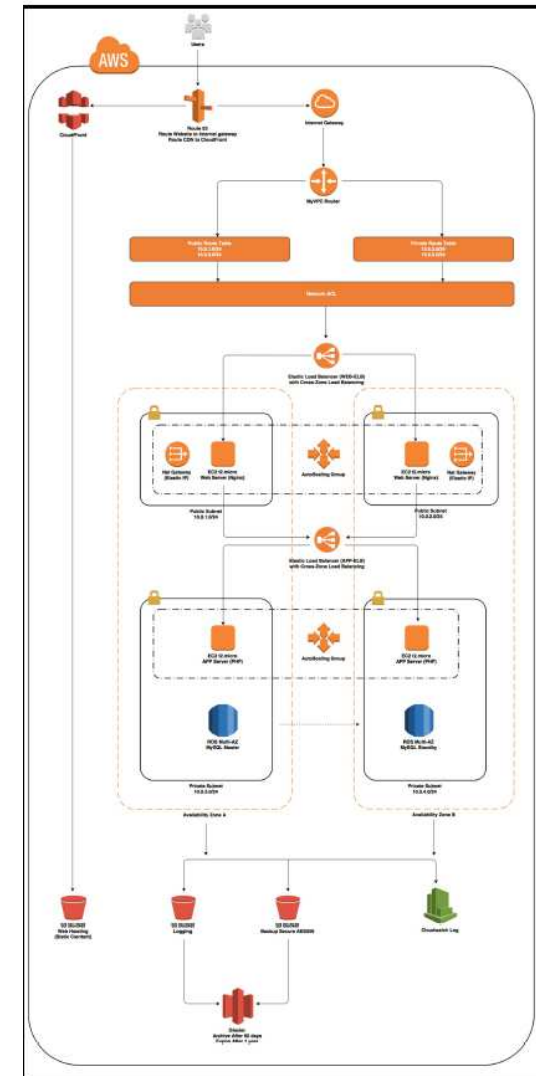


# **Plantillas : AWS Cloudformation**

- Práctica : AWS Cloudformation
- Analizar y desplegar vpc.json
- ¿Que es lo que crea?
- Esta plantilla como base esta muy bien!!!!!!!!!!!!
- Pero todo en un único fichero

# Plantillas : AWS Cloudformation

- Práctica : AWS Cloudformation
- **Plantilla multifichero** para la creación de una infraestructura para una aplicación web para producción
  - IAM
  - VPC
  - EC2
  - ELB
  - AutoScaling
  - CloudFront
  - RDS
  - S3
  - Cloudwatch
  - Route53
  - Security Group y ACL



# **Plantillas : AWS Cloudformation**

- Práctica : AWS Cloudformation
- Para realizar la práctica es necesario
  - Una cuenta AWS con acceso a los recursos anteriores
    - Starter Account , todos menos IAM
  - Tener un dominio en Route53
    - Domeinua.eus
  - Tener un ACM (AWS Certificate Manager)

# Plantillas : AWS Cloudformation

- Práctica : AWS Cloudformation
- **Plantilla multifichero** para la creación de una infraestructura para una aplicación web para producción
- Conceptos a desplegar
  - VPC con subredes privadas y públicas
  - EC2 con HA desplegando en dos AZ un grupo de autoescalado
  - Nat gateways para gestionar el trafico saliente
  - Un ELB para gestionar el tráfico entrante
  - Logging centralizado con Cloudwatch
  - S3 para hosting estático y logging de ELB
  - Base de datos multi AZ



# Plantillas : AWS Cloudformation

- Práctica : AWS Cloudformation
- **Plantilla multifichero** para la creación de una infraestructura para una aplicación web para producción
- Se gestionan tres tipos de entornos de despliegue
  - Test
  - Dev
  - Prod

# Plantillas : AWS Cloudformation

- Pasos para el despliegue
- Clonar repositorio con el IaC
  - <https://github.com/awsstar/CloudFormation-WebApp-Architecture>
- Subir las plantillas a S3
  - `aws s3 sync ./ s3://mgep-master-automation/cloudformation`

upload: infrastructure/webapp-vpc.yaml to s3://mgep-master-automation/cloudformation/infrastructure/webapp-vpc.yaml

upload: ./master.yaml to s3://mgep-master-automation/cloudformation/master.yaml

- Obtener id del Certificado relacionado con vuestro host zone

```
$aws acm list-certificates
```

```
{
  "CertificateSummaryList": [
    {
      "CertificateArn": "arn:aws:acm:eu-west-1:420693608596:certificate/cad9edea-b5e6-42c6-b993-e1ea19906e4e",
      "DomainName": "www.jaagirre.eus"
    }
  ]
}
```

# Plantillas : AWS Cloudformation

- Pasos para el despliegue
- Modificar master.yaml con nuestros datos
  - Parametros
    - IP de acceso a recursos
    - Clave de acceso a recursos
    - Bucket de s3
    - Dominio
  - Mappings
    - Dominio en los tres entornos de despliegue
    - Y AMI, transición de almacenamiento y certificado para las diferentes regiones

# Plantillas : AWS Cloudformation

- Recursos de master.yaml
  - Aquí se realiza elegantemente la división en ficheros con el tipo de recurso "AWS::CloudFormation::Stack"
  - Se definen los siguientes recursos/plantillas
    - MyIAMRole:
      - Type: "AWS::CloudFormation::Stack"
    - MyS3Bucket:
      - Type: "AWS::CloudFormation::Stack"
    - MyVPC:
      - Type: "AWS::CloudFormation::Stack"
    - MySecurityGroup:
      - Type: "AWS::CloudFormation::Stack"
    - MyRDS:
      - Type: "AWS::CloudFormation::Stack"
    - MyAPPELB:
      - Type: "AWS::CloudFormation::Stack"
    - MyAPPAutoScaling:
      - Type: "AWS::CloudFormation::Stack"
    - MyWEBELB:
      - Type: "AWS::CloudFormation::Stack"
    - MyWEBAutoScaling:
      - Type: "AWS::CloudFormation::Stack"
    - MyCloudWatch:
      - Type: "AWS::CloudFormation::Stack"
    - MyDNS:
      - Type: "AWS::CloudFormation::Stack"
- Salidas de master.yaml
  - HTTP Endpoint
  - Certificados SSL

```
MyIAMRole:
  Type: "AWS::CloudFormation::Stack"
  Properties:
    TemplateURL: !Sub "${PMTemplateURL}/webapp-iam.yaml"
    TimeoutInMinutes: '5'
  Parameters:
    PMServerEnv: !Ref "AWS::StackName"
```

# Plantillas : AWS Cloudformation

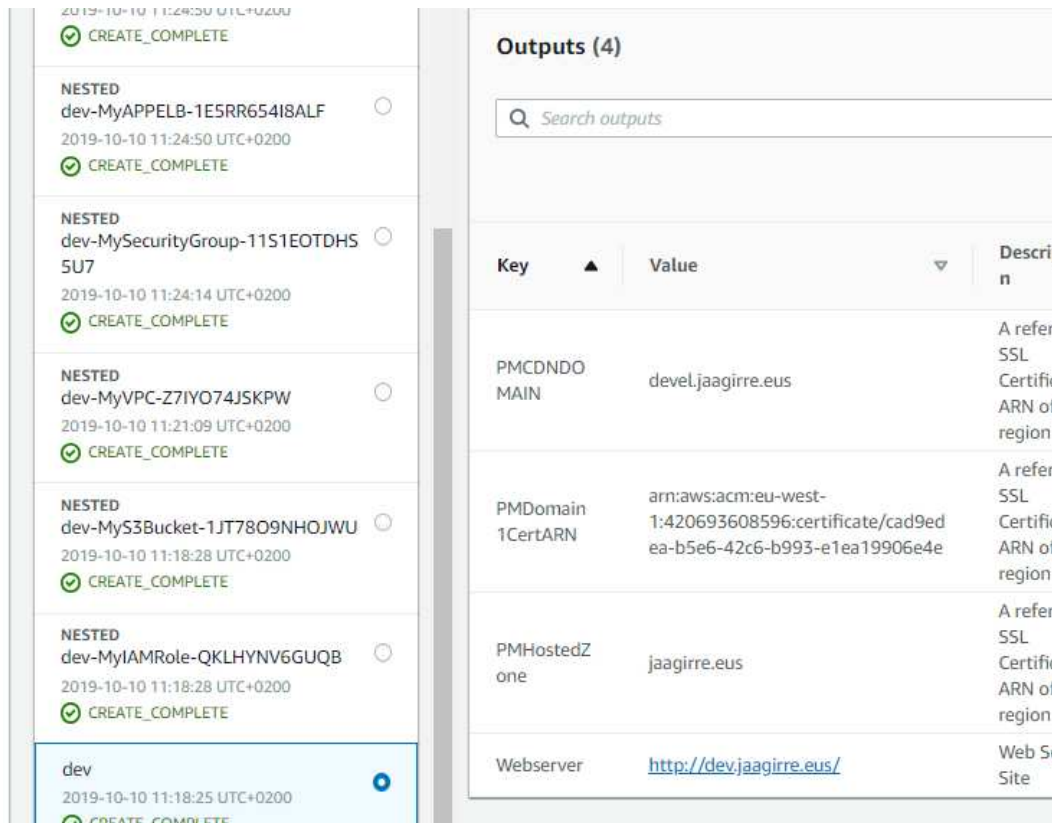
- Recursos de master.yaml
- Establecer correctamente el bucket donde se almacenan los templates
- Ejecutar

The screenshot displays the AWS CloudFormation console. On the left, a list of stacks is shown, including nested stacks. On the right, the 'Events' tab is selected, showing a table of events for the selected stack.

| Timestamp                    | Logical ID       | Status             |
|------------------------------|------------------|--------------------|
| 2019-10-10 09:12:18 UTC+0200 | MyAPPAutoScaling | CREATE_IN_PROGRESS |
| 2019-10-10 09:12:16 UTC+0200 | MyAPPELB         | CREATE_COMPLETE    |
| 2019-10-10 09:12:06 UTC+0200 | MyAPPELB         | CREATE_IN_PROGRESS |
| 2019-10-10 09:12:05 UTC+0200 | MyRDS            | CREATE_IN_PROGRESS |
| 2019-10-10 09:12:05 UTC+0200 | MyAPPELB         | CREATE_IN_PROGRESS |
| 2019-10-10 09:12:05 UTC+0200 | MyRDS            | CREATE_IN_PROGRESS |
| 2019-10-10 09:12:03 UTC+0200 | MySecurityGroup  | CREATE_COMPLETE    |
| 2019-10-10 09:11:29 UTC+0200 | MySecurityGroup  | CREATE_IN_PROGRESS |
| 2019-10-10 09:11:29 UTC+0200 | MySecurityGroup  | CREATE_IN_PROGRESS |

# Plantillas : AWS Cloudformation

- master.yaml
- Una vez desplegado analizar los desplegado junto con cada una de las plantillas



| Key               | Value                                                                               | Description                                          |
|-------------------|-------------------------------------------------------------------------------------|------------------------------------------------------|
| PMCDNDO MAIN      | devel.jaagirre.eus                                                                  | A reference to the SSL Certificate ARN of the region |
| PMDomain 1CertARN | arn:aws:acm:eu-west-1:420693608596:certificate/cad9edea-b5e6-42c6-b993-e1ea19906e4e | A reference to the SSL Certificate ARN of the region |
| PMHostedZone      | jaagirre.eus                                                                        | A reference to the SSL Certificate ARN of the region |
| Webservice        | <a href="http://dev.jaagirre.eus/">http://dev.jaagirre.eus/</a>                     | Web Site                                             |



This is App Server

```

PHP Version 5.3.29

System      Linux ip-10-0-4-194 4.1.10-x86_64
Build Date  May 12 2015 22:43:14
Configure Command './configure' '--build=x86_64-amazon-linux' '--target=x86_64-amazon-linux' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--includedir=/usr/include'
  
```

# **Plantillas : AWS Cloudformation**

- Una vez desplegado analizar todos los ficheros y entender la plantilla
  - iamrole.yaml
  - S3.yaml
  - Vpc.yaml
  - Securitygroup.yaml
  - appElb.yaml
  - Rds.yaml
  - appAutoscaling.yaml
  - webELB.yaml
  - Webautoscalling.yaml
  - Dns.yaml
  - Cloudwatch.yaml

# Plantillas : AWS Cloudformation

- **iamrole.yaml**
- Recibe como parámetro el nombre del stack, que especifica la fase de ciclo de vida
- Crea un dos roles
  - **IAMS3CW** : Role para recursos EC2 con políticas full Access a S3 y Cloudwatch preestablecidas
    - Crea una identidad/perfil IAM para asociarle el rol anterior
  - **IAMVPCLog** : Role con politica manual para crear logs para servicios flow de VPC
    - Crea una identidad/perfil IAM para asociarle el rol anterior
- Output: Nombre de los perfiles IAM creados y el ARN del role VPC

```
IAMS3CW:
  Type: "AWS::IAM::Role"
  Properties:
    ManagedPolicyArns:
      - "arn:aws:iam::aws:policy/CloudWatchFullAccess"
      - "arn:aws:iam::aws:policy/AmazonS3FullAccess"
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        -
          Effect: "Allow"
          Principal:
            Service:
              - "ec2.amazonaws.com"
          Action:
            - "sts:AssumeRole"
    Path: "/"
```

```
IAMS3CWInstanceProfile:
  Type: "AWS::IAM::InstanceProfile"
  Properties:
    Path: "/"
    Roles:
      -
        Ref: "IAMS3CW"
```



# Plantillas : AWS Cloudformation

- S3.yaml
- Parametros
  - El tipo transición de almacenamiento dependiendo de la región
  - La fase de ciclo de vida
- Crea 2 buckets privados y 1 público
  - S3Backup : Privado
    - Con política que obliga a encriptación AES256
    - Subida
  - S3Logging : Privado
    - Sin encriptación
    - Subida
  - S3CloudFront : Publico para CDN
    - Tipo website
    - Lectura
- Outputs
  - Exporta nombre de dominio bucket CDN

```
S3Backup:
  Type: "AWS::S3::Bucket"
  Properties:
    AccessControl: "Private"
    VersioningConfiguration:
      Status: "Enabled"
    LifecycleConfiguration:
      Rules:
        - Id: "MyBackupArchive"
          Status: "Enabled"
          ExpirationInDays: "365" # Complete Disposal/Deletion of Data
          Transition:
            TransitionInDays: "60" # Move Data from S3 bucket to Archive
            StorageClass: !Ref "PMRegionASTorage"
          DeletionPolicy: "Retain"

S3BackupPolicy:
  Type: "AWS::S3::BucketPolicy"
  Properties:
    Bucket: !Ref "S3Backup"
    PolicyDocument:
      Statement:
        - Sid: "DenyUnEncryptedObjectUploads"
          Effect: "Deny"
          Principal:
            AWS: "*"
          Action: "s3:PutObject"
          Resource: !Join ["", ["arn:aws:s3:::", !Ref "S3Backup", "/*"]]
          Condition:
            StringNotEquals:
              s3:x-amz-server-side-encryption: "AES256"
```

# Plantillas : AWS Cloudformation

- Vpc.yaml
  - Role para logs VPC
  - Crea un VPC
    - FlowLog con permisos
    - 2 subredes publicas
    - 2 subreds privadas
    - 1 Gateway de internet
    - 1 Attachement de Gateway a VPC
    - 2 Servidores NAT con EIP
    - 4 tablas de enrutamiento
    - 2 rutas para las subredes publicas hacia internet
    - 2 rutas para las subredes privadas hacia su NAT
    - 4 Asociaciones de rutas a tablas subredes
    - 1 acl y 4 asociaciones a subredes
      - Acls por defecto (allow )
- Outputs
  - Exporta VPC

# Plantillas : AWS Cloudformation

- Securitygroup.yaml
- Parametro
  - VPC Id
  - Id de ACL de red sin configurar previamente
- Recursos
  - 5 grupos de seguridad
    - Se relacionaran a posteriori con los servidores Ec2 y otros recursos
    - Y se asocian al VPC
    - RDS 3306
    - Application ELB 9000
    - Application server 22, 9000
    - WEB ELB icmp, 80 , 443
    - WEB server 22, 443, 80
  - 5 Entradas de ACL de entrada : 22, 80, 443 , 1024-65535 , ICMP
  - 3 Entradas de ACL de salida hacia : 80, 443 , 22 , 1024-65535, ICMP
    - Se relacionan con el ACL que se relacionara con todas las subredes
- Outputs
  - Se exporta al grupo de seguridad par RDS

# **Plantillas : AWS Cloudformation**

- appElb.yaml
- Parámetros
  - Grupo de seguridad
  - Subredes en las cuales lanzar las instancias
  - Bucket de logging
- Recursos
  - 1 Loadbalancer con crosszone
    - Logging de accesos
    - Healthcheck
    - Puerto 9000
- Output
  - Nombre DNS del loadbalancer

# **Plantillas : AWS Cloudformation**

---

- Rds.yaml
- Parámetros
  - Username, password, subnet , databasename
  - ELB security group
- Recursos
  - Instancia DB
  - Multiaz dependiendo del tipo de despliegue

# Plantillas : AWS Cloudformation

- appAutoscaling.yaml
- Parametros
  - Par de claves EC2
  - Subredes
  - Grupo de seguridad de Hosts
  - Referencia al nombre de dominio
  - **Referencia al App ELB**
  - Referencia al perfil/role para EC2
  - Datos de autoescalado
- Recursos
  - 1 configuración de autoescalado con la definicion del tipo de servidor con un Role
    - User data: htop, php , mcrypt, telnet , git , Python , pip , aws , nginx, aws-cfn-bootstrap, nginx, index.php
  - 1 Grupo de autoescalado relacionada con la configuración de lanzamiento y el ELB , 1 política de escalado UP y otra DOWN
  - 1Condicion de espera
- Outputs

```
sed -ie 's/127.0.0.1:9000/9000/g' /etc/php-fpm.d/www.conf  
sed -ie 's/listen.allowed_clients;/listen.allowed_clients/g' /etc/php-fpm.d/www.conf  
sed -ie 's/user = apache/user = nginx/g' /etc/php-fpm.d/www.conf  
sed -ie 's/group = apache/group = nginx/g' /etc/php-fpm.d/www.conf
```

# **Plantillas : AWS Cloudformation**

- webELB.yaml
- Parametros
  - Subred publica , bucket de backup, bucket de logs certificado
- Recursos
  - 1 loadbalancer
    - 80 , 443
    - Certificado SSL
    - Cross-zone
    - Grupo de seguridad
    - Subred
- Outputs

# Plantillas : AWS Cloudformation

- Webautoscalling.yaml
- Parámetros
  - Grupo de seguridad , subred pública, nombre de dominio
  - Dirección al balanceador de carga de aplicación
  - Balanceador de carga web
- Recursos
  - 1 Configuración de despliegue
    - Nginx , htop
  - 1 Condición de espera, 1 Grupo de autoescalado con la configuración anterior
    - Configuraciones de escalado
- Outputs

```
... ##
... # Upstream Server
... ##
... upstream php {
...     server ${PMAPPLoadBalancerUrl}:9000;
... }
... ##
```



# Plantillas : AWS Cloudformation

- Dns.yaml

|                                                                                   |                   |   |                                                |    |
|-----------------------------------------------------------------------------------|-------------------|---|------------------------------------------------|----|
|  | dev.jaagirre.eus. | A | ALIAS dev-webelb-1514518074.eu-west-1.elb.amaz | No |
|-----------------------------------------------------------------------------------|-------------------|---|------------------------------------------------|----|

- Parámetros
  - información para crear el registro del ALIAS
    - Nombre de dominio
    - Referencia al hosted zone
    - Nombre de dominio del ELB
    - ID del Hosted zone del WEB ELB

- Recursos
  - 1 RecordSetGroup
- Outputs

```
Route53:
  Type: "AWS::Route53::RecordSetGroup"
  Properties:
    HostedZoneName: !Sub "${PMHostedZone}."
    Comment: "Zone apex alias targeted to myELB LoadBalancer."
    RecordSets:
      - Name: !Sub "${PMWEBDOMAIN}."
        Type: 'A'
        AliasTarget:
          HostedZoneId: !Ref "PMWEBLBHostedZoneId"
          DNSName: !Ref "PMWEBLBDNSName"
```

# Plantillas : AWS Cloudformation

- Cloudwatch.yaml
- Parámetros
  - Referencias a las políticas de autoescalado UP/DOWN, a los grupos de autoescalado para definir las **alarmas** par activar los autoeasaldo
- Recursos
  - 4 Alarmas relacionadas con sus políticas de escalado y grupo de escalado
- Outputs

```
# App Server Alarm Low Load
AppCPUAlarmLow:
  Type: "AWS::CloudWatch::Alarm"
  Properties:
    AlarmDescription: "Scale-down if CPU < 25% for 5 minutes"
    MetricName: "CPUUtilization"
    Namespace: "AWS/EC2"
    Statistic: "Average"
    Period: "300"
    EvaluationPeriods: "1"
    Threshold: "25"
    AlarmActions:
      - Ref: "PMAPPServerScaleDownPolicy"
    Dimensions:
      - Name: "AutoScalingGroupName"
      - Value:
          Ref: "PMAAppScalingGroup"
    ComparisonOperator: "GreaterThanThreshold"
```

# **Plantillas : AWS Cloudformation**

- Ejemplos de cloudformation

[https://docs.aws.amazon.com/es\\_es/AWSCloudFormation/latest/UserGuide/sample-templates-applications-eu-west-1.html](https://docs.aws.amazon.com/es_es/AWSCloudFormation/latest/UserGuide/sample-templates-applications-eu-west-1.html)