AWS - CloudFormation

mardi 21 février 2023 09:31

Formateur : Mohamed AIJJOU

https://docs.aws.amazon.com/fr_fr/AWSCloudFormation/latest/UserGuide/Welcome.html

https://aws.amazon.com/fr/cloudformation/

https://fr.wikipedia.org/wiki/Infrastructure_as_code

Infrastructure as code

11 langues

Article Discussion Lire Modifier Modifier le code Voir l'historique

Pour les articles homonymes, voir IAC.

L'***Infrastructure as code*** (**IaC**) (littéralement : « infrastructure en tant que code ») est un ensemble de mécanismes permettant de gérer, par des fichiers descripteurs ou des scripts (code informatique), une infrastructure (informatique) virtuelle[1,2].

Initialement dédié aux machines virtuelles (également nommées « Instances »), l'évolution des offres dans le domaine de la virtualisation a rendu possible la gestion d'une infrastructure à part entière, de l'instance au réseau, incluant entre autres la gestion du service DNS, du « Load-Balancing », des sous-réseaux et des groupes de sécurité[3].

Souvent plébiscité dans le cadre du cloud computing, l'Infrastructure as code offre aux développeurs la possibilité d'automatiser leurs déploiements de manière à éviter les tâches manuelles ou encore de devoir écrire d'eux-mêmes les appels aux interfaces de programmation. Cette technologie constitue une réponse aux besoins des entreprises en termes de mise à l'échelle des applications axée sur l'automatisation et la simplification de l'infrastructure de projets informatiques. De manière générale, l'Infrastructure as code s'inscrit dans la mouvance plus générale du DevOps qui a pour objectif d'unifier le développement logiciel et l'administration système.

AWS CloudFormation est un service qui vous permet de modéliser et de configurer vos ressources AWS de sorte que vous puissiez passer moins de temps à gérer ces ressources et consacrer plus de temps à vos applications exécutées dans AWS. Vous créez un modèle qui décrit toutes les ressources AWS que vous voulez (telles que des instances Amazon EC2 ou des instances de base de données Amazon RDS), et CloudFormation s'occupe de leur allocation et de leur configuration. Vous n'avez pas besoin de créer et de configurer individuellement les ressources AWS ni de déterminer leurs dépendances. CloudFormation se charge de tout. Les scénarios suivants montrent comment CloudFormation peut vous aider.

Fonctionnement

AWS CloudFormation vous permet de modéliser, de provisionner et de gérer les ressources AWS et tierces en traitant l'infrastructure en tant que code.

Cas d'utilisation

Gérer l'infrastructure avec DevOps

Automatiser, tester et déployer des modèles d'infrastructure avec des automatisations d'intégration et de livraison continues (CI/CD).

Mettre à l'échelle des piles de production

Exécuter tout ce que vous voulez, d'une seule instance Amazon Elastic Compute Cloud (EC2) à une application multi-régions complexe.

Partager les bonnes pratiques

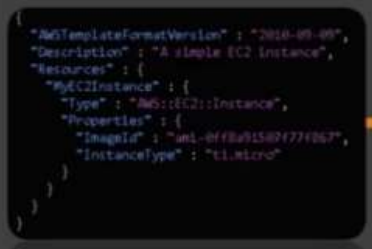
Définir facilement un sous-réseau Amazon Virtual Private Cloud (VPC) ou des services d'approvisionnement comme AWS OpsWorks ou Amazon Elastic Container Service (ECS).

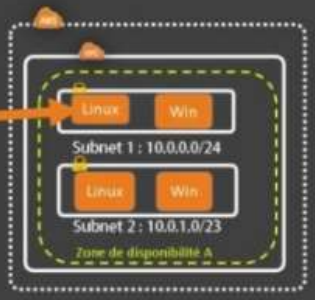
Mots clés :

- Simplifie
- Tâches répétitives
- provisionnement

- AWS Management Console
- AWS CLI
- AWS SDK / API

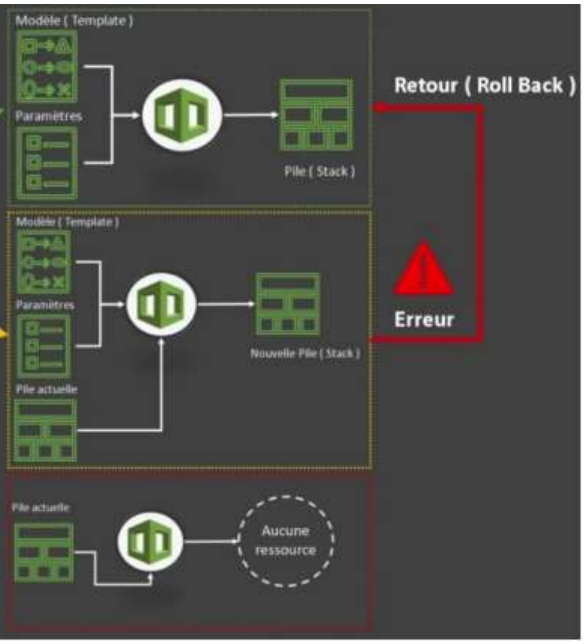
- Modèle (Template)

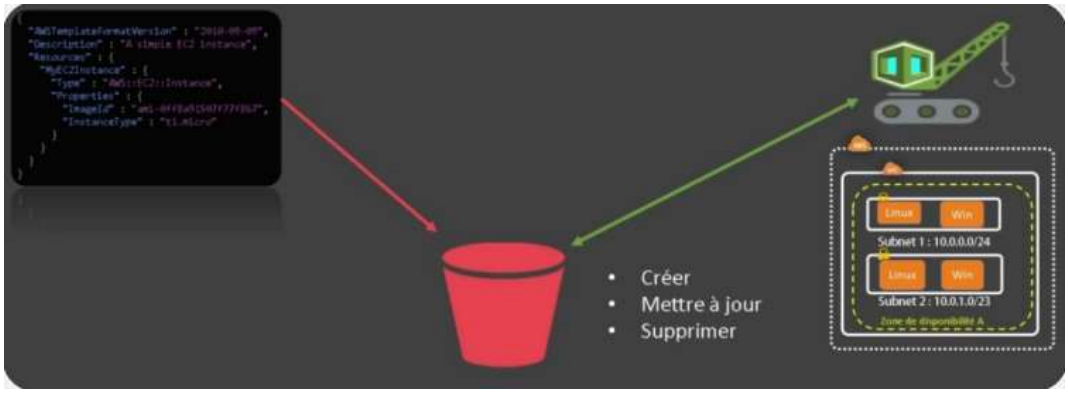


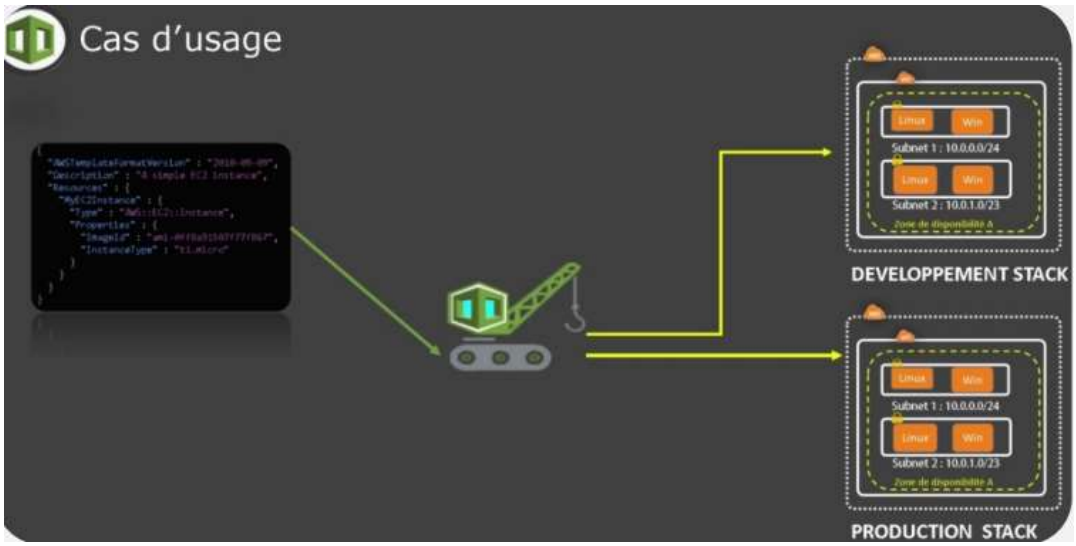


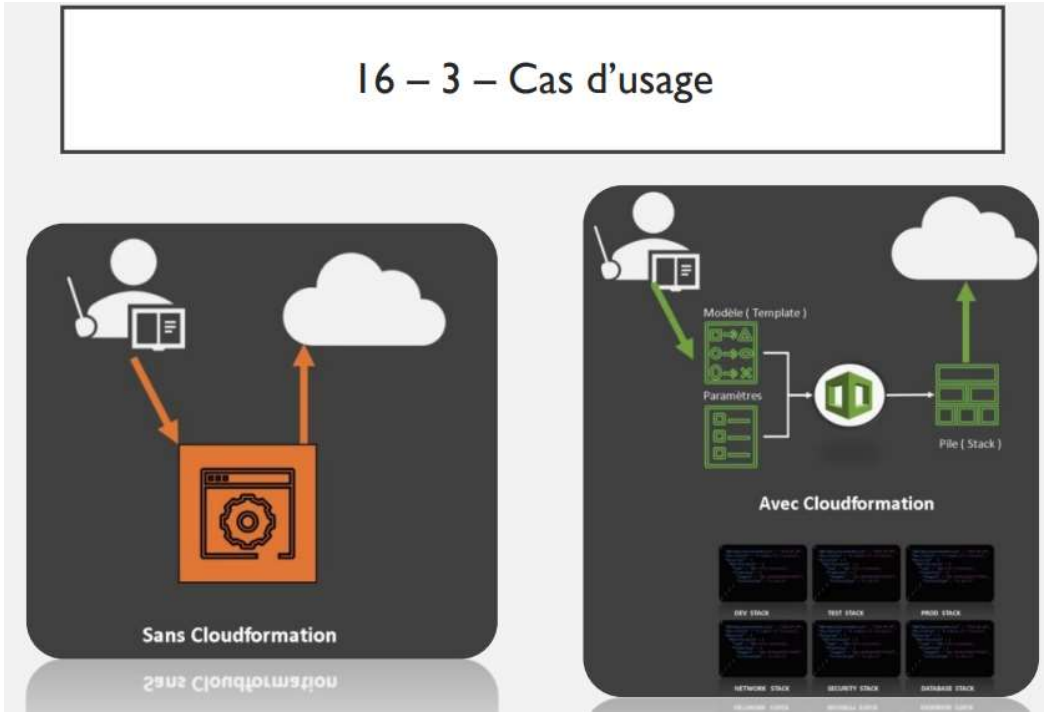
Fonctionnalités

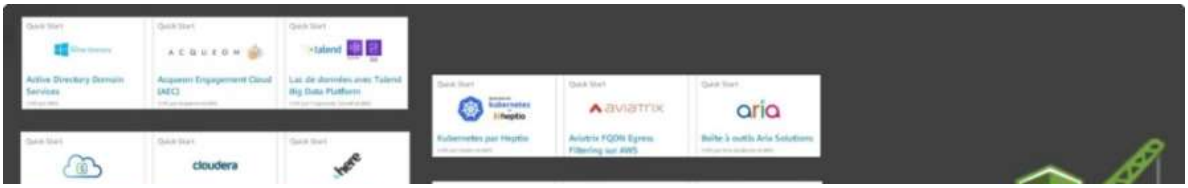
- Créer
- Mettre à jour
- Supprimer




Linux
Win
Subnet 1 : 10.0.0.0/24
Linux
Win
Subnet 2 : 10.0.1.0/23
Zone de disponibilité A
• Créer
• Mettre à jour
• Supprimer


Cas d'usage
Linux
Win
Subnet 1 : 10.0.0.0/24
Linux
Win
Subnet 2 : 10.0.1.0/23
Zone de disponibilité A
DEVELOPPEMENT STACK
Linux
Win
Subnet 1 : 10.0.0.0/24
Linux
Win
Subnet 2 : 10.0.1.0/23
Zone de disponibilité A
PRODUCTION STACK


16 – 3 – Cas d'usage
Sans Cloudformation
Modèle (Template)
Paramètres
Pile (Stack)
Avec Cloudformation


cloudera

Exemple de template :

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html

Exemple de fichier de configuration :

YAML

```yaml
Type: AWS::EC2::Instance
Properties:
  AdditionalInfo: String
  Affinity: String
  AvailabilityZone: String
  BlockDeviceMappings:
    - BlockDeviceMapping
  CpuOptions:
    CpuOptions
  CreditSpecification:
    CreditSpecification
  DisableApiTermination: Boolean
  EbsOptimized: Boolean
  ElasticGpuSpecifications:
    - ElasticGpuSpecification
  ElasticInferenceAccelerators:
    - ElasticInferenceAccelerator
  EnclaveOptions:
    EnclaveOptions
  HibernationOptions:
    HibernationOptions
  HostId: String
  HostResourceGroupArn: String
  IamInstanceProfile: String
  ImageId: String
  InstanceInitiatedShutdownBehavior: String
```

Les ressources présente dans ce fichier peuvent faire écho à des ressources existantes : ici on voit que le fichier de configuration propose les options "**Subnet**" et "**SecurityGr** ressources à utiliser dans la création de notre **EC2** :

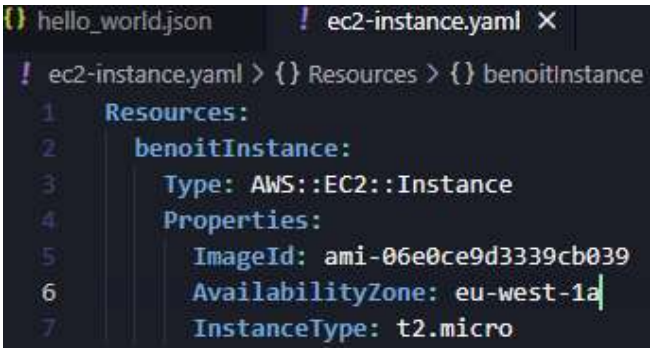
```yaml
SecurityGroupIds:
  - String
SecurityGroups:
  - String
SourceDestCheck: Boolean
SsmAssociations:
```

```
  - SsmAssociation
SubnetId: String
Tags:
  - Tag
Tenancy: String
UserData: String
Volumes:
  - Volume
```

TP : générer un 1er fichier de configuration .yaml et le déployer

Pour nous faciliter notre développement, nous pouvons installer l'addon **CloudFormation** sur VSCode :

Capture d'écran de notre script yaml :

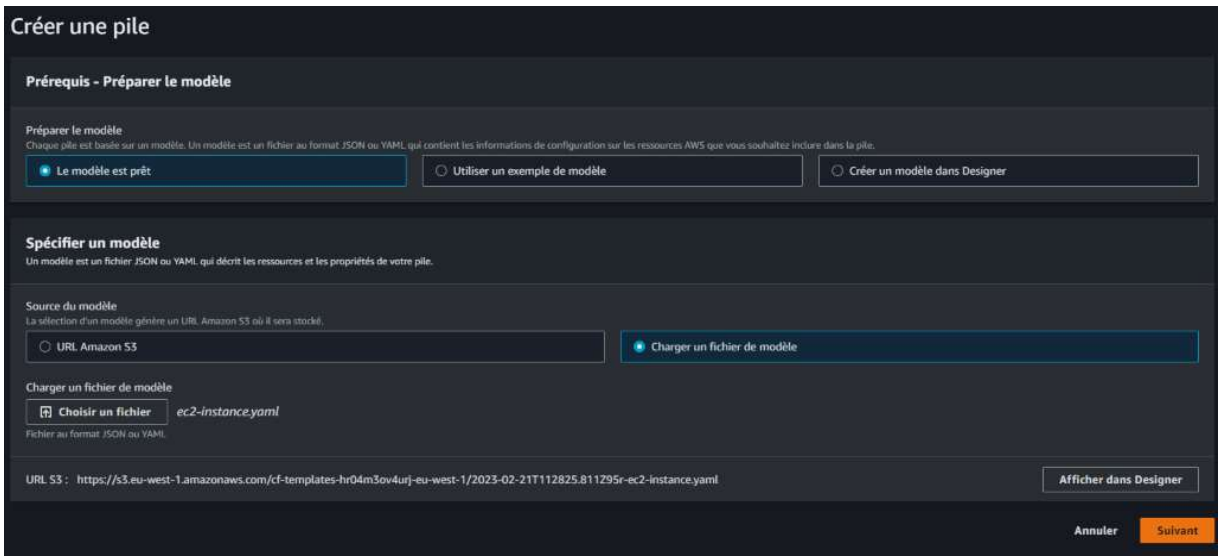


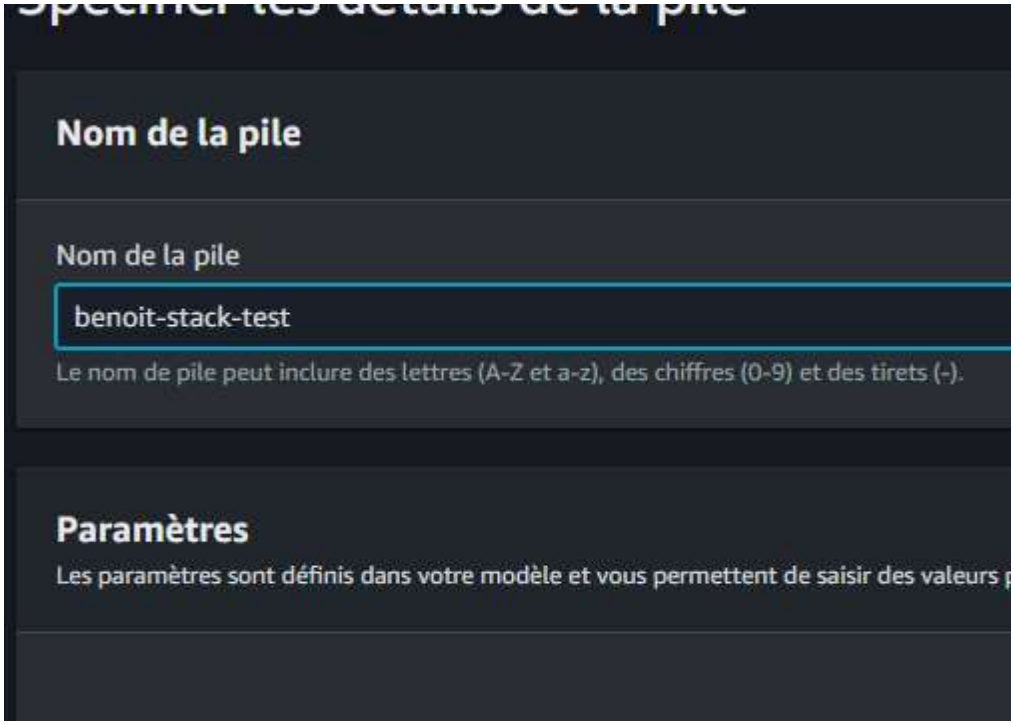
Attention : l'AMI est différente en fonction des régions, bien s'assurer que nous sommes sur la bonne région quand on récupére l'AMI

De même, bien faire attention pour la zone de disponibilité : Il est important de définir une zone d'une région, par exemple: **eu-west-1a** ou **eu-west-2-a**

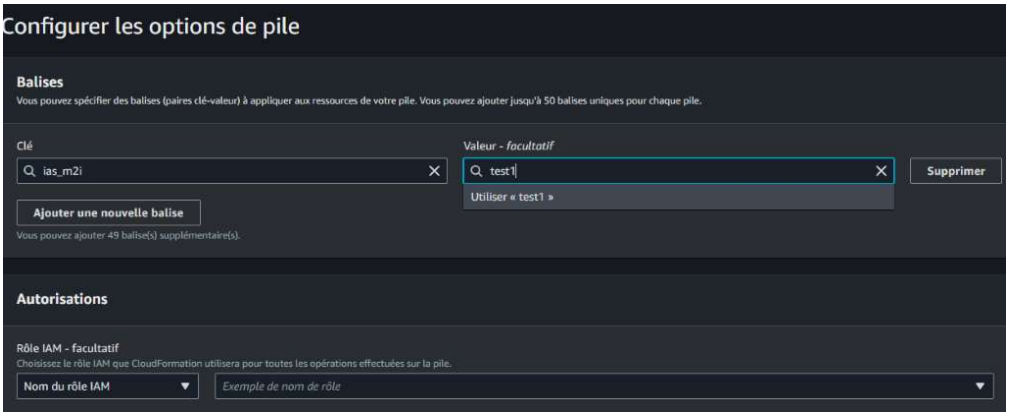
Pour ce build, on gardera uniquement les propriétés **ImageId**, **AvailabilityZone** et **InstanceType**

On va tester la création d'une stack avec notre build en rajoutant notre fichier :

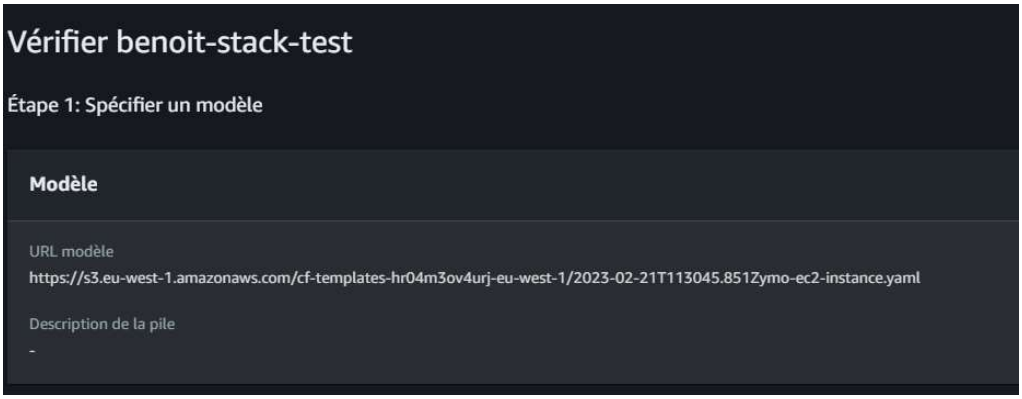




On va configurer notre stack : les parties **clé-valeur sont obligatoire**



On peut également vérifier notre build :

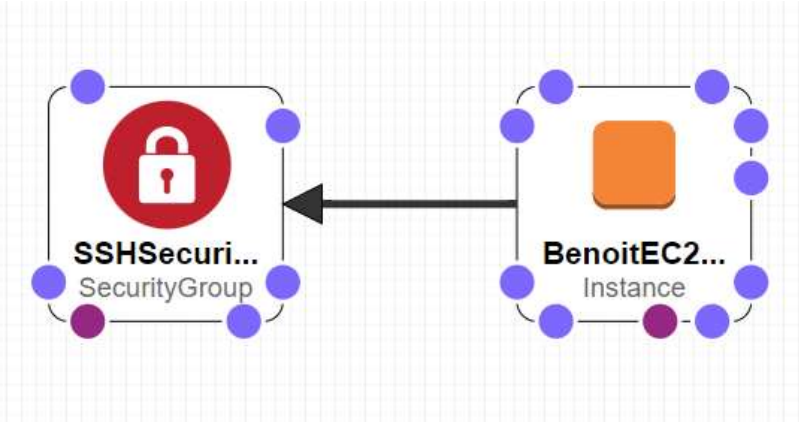


Second TP : création d'un script yaml qui génère une instance ec2 et un groupe se sécurité :

```
        SecurityGroupIds:
          - default
          - !Ref SSHSecurityGroupBenoit
    SSHSecurityGroupBenoit:
      Type: AWS::EC2::SecurityGroup
      Properties:
        GroupName: gs-benoit-iac
        GroupDescription: groupe de securite ssh pour exercice iac
        SecurityGroupIngress:
          - IpProtocol: tcp
            FromPort: '22'
            ToPort: '22'
            CidrIp: 0.0.0.0/0
```

Le **!Ref** permet de faire référence à des ressources présente dans le fichier de configuration : syntaxe : **!Ref "nom_ressources"**

On peut vérifier dans le designer si tout a l'air bon :



Build de la stack :

Événements (8)

Rechercher des événements

Horodatage	ID logique	Statut	Motif du statut
21-02-2023 14:19:56 UTC+0100	benoit-ecs-gs-ssh	CREATE_COMPLETE	-
21-02-2023 14:19:54 UTC+0100	BenoitEC2Instance	CREATE_COMPLETE	-
21-02-2023 14:19:02 UTC+0100	BenoitEC2Instance	CREATE_IN_PROGRESS	Resource creation Initiated
21-02-2023 14:19:00 UTC+0100	BenoitEC2Instance	CREATE_IN_PROGRESS	-
21-02-2023 14:18:59 UTC+0100	SSHSecurityGroupBenoit	CREATE_COMPLETE	-
21-02-2023 14:18:58 UTC+0100	SSHSecurityGroupBenoit	CREATE_IN_PROGRESS	Resource creation Initiated
21-02-2023 14:18:53 UTC+0100	SSHSecurityGroupBenoit	CREATE_IN_PROGRESS	-
21-02-2023 14:18:49 UTC+0100	benoit-ecs-gs-ssh	CREATE_IN_PROGRESS	User Initiated

Tp : création d'un script yaml qui génère une instance ec2, deux groupes se sécurité et une Elastic IP :

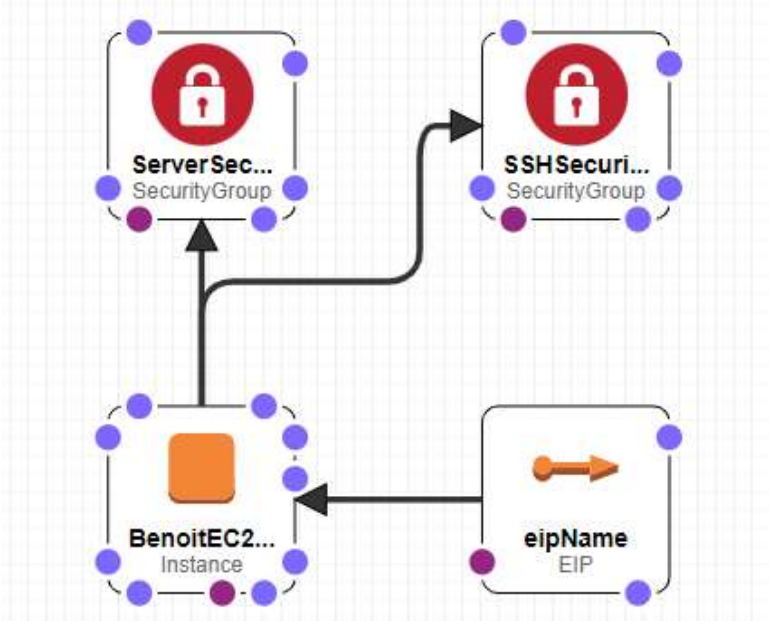
Code source de l'exercice :

ec2-instance-gs-eip.yaml > { } Resources > { } BenoitEC2Instance > { } Properties > [ ] Secu

```
Resources:
  BenoitEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-06e0ce9d3339cb039
      InstanceType: t2.micro
      AvailabilityZone: eu-west-1a
      SecurityGroupIds:
        - !Ref SSHSecurityGroupBenoit
        - !Ref ServerSecurityGroup
      Tags:
      - Key: Name
        Value: benoit-instance
```

```yaml
      Value: benoit-instance
  SSHSecurityGroupBenoit:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: gs-benoit-iac
      GroupDescription: groupe de securite ssh pour exercice iac
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '22'
          ToPort: '22'
          CidrIp: 0.0.0.0/0
  ServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: gs-benoit-web
      GroupDescription: groupe de securite web
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
          ToPort: '80'
          CidrIp: 0.0.0.0/0
  eipName:
    Type: AWS::EC2::EIP
    Properties:
      InstanceId: !Ref BenoitEC2Instance
```

Vérification du corde de designer :



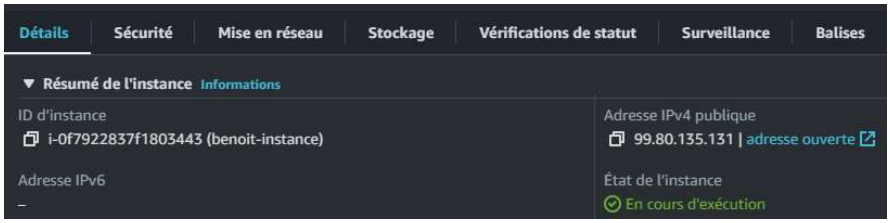
Build de la stack :

Événements (14)

Rechercher des événements

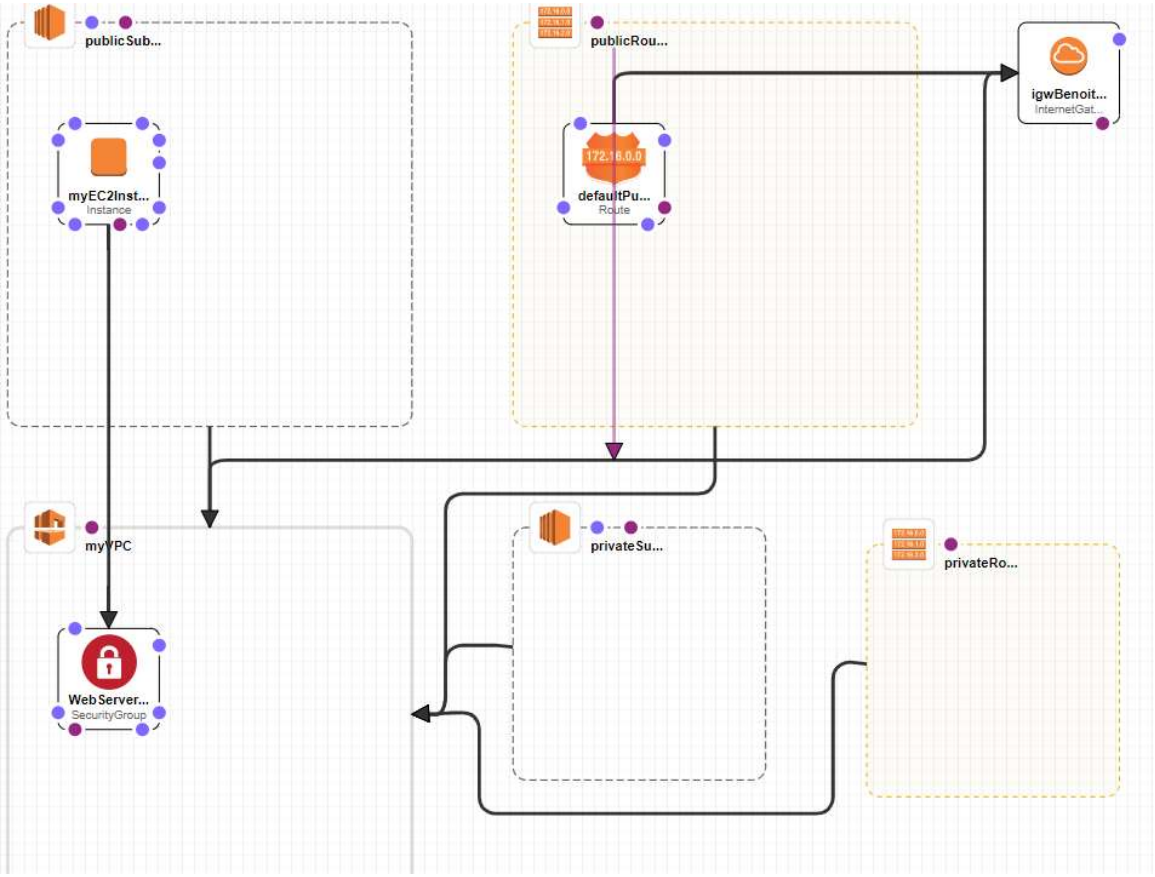
Horodatage	ID logique	Statut	Motif du statut
21-02-2023 14:48:51 UTC+0100	benoit-stack-ec2-gs2-eip	CREATE_COMPLETE	-
21-02-2023 14:48:50 UTC+0100	eipName	CREATE_COMPLETE	-
21-02-2023 14:48:33 UTC+0100	eipName	CREATE_IN_PROGRESS	Resource creation Initiated
21-02-2023 14:48:31 UTC+0100	eipName	CREATE_IN_PROGRESS	-
21-02-2023 14:48:30 UTC+0100	BenoitEC2Instance	CREATE_COMPLETE	-
21-02-2023 14:47:59 UTC+0100	BenoitEC2Instance	CREATE_IN_PROGRESS	Resource creation Initiated
21-02-2023 14:47:57 UTC+0100	BenoitEC2Instance	CREATE_IN_PROGRESS	-
21-02-2023 14:47:56 UTC+0100	ServerSecurityGroup	CREATE_COMPLETE	-
21-02-2023 14:47:55 UTC+0100	SSHSecurityGroupBenoit	CREATE_COMPLETE	-
21-02-2023 14:47:55 UTC+0100	ServerSecurityGroup	CREATE_IN_PROGRESS	Resource creation Initiated
21-02-2023 14:47:55 UTC+0100	SSHSecurityGroupBenoit	CREATE_IN_PROGRESS	Resource creation Initiated
21-02-2023 14:47:50 UTC+0100	ServerSecurityGroup	CREATE_IN_PROGRESS	-
21-02-2023 14:47:50 UTC+0100	SSHSecurityGroupBenoit	CREATE_IN_PROGRESS	-
21-02-2023 14:47:47 UTC+0100	benoit-stack-ec2-gs2-eip	CREATE_IN_PROGRESS	User Initiated

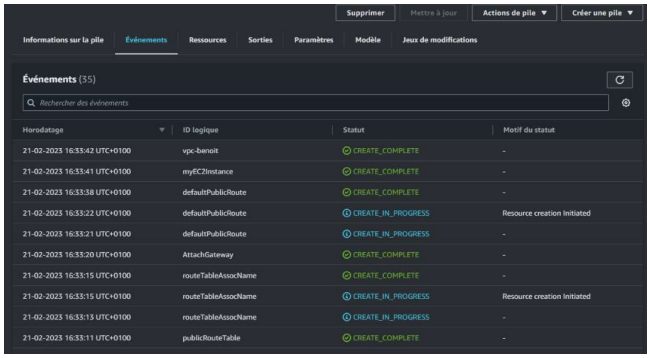
Instance : i-0f7922837f1803443 (benoit-instance)



TP : création infrastructure VPC + EC2

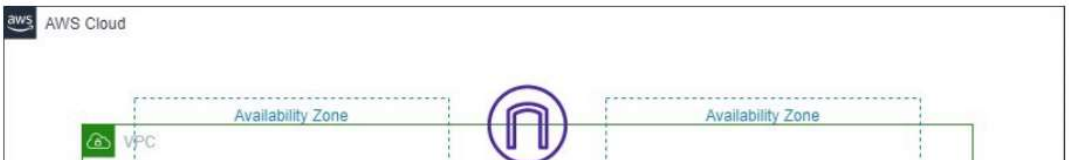
https://docs.aws.amazon.com/fr_fr/AWSCloudFormation/latest/UserGuide/mappings-section-structure.html

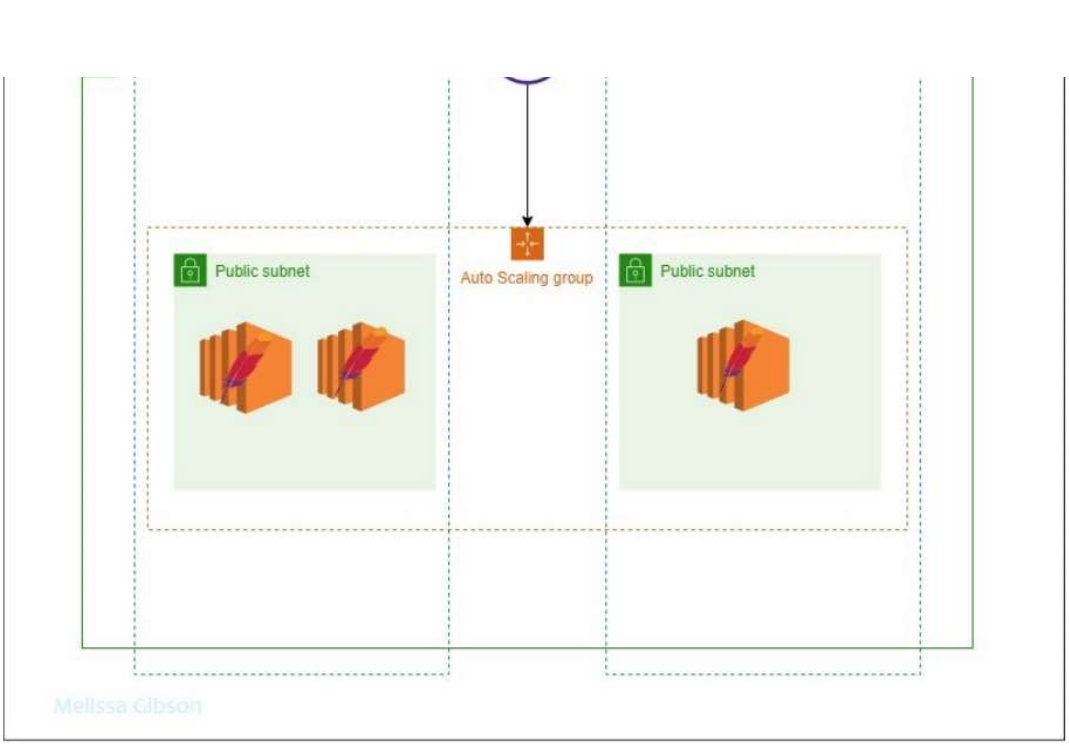




Code source de ce TP disponible dans le dossier "Script" :

TP : réaliser cette infrastructure





Correction :

https://towardsaws.com/creating-an-auto-scaling-group-in-aws-cloudformation-761b0f8c6364