

AWS – Pipeline CI/CD

jeudi 23 février 2023 14:02

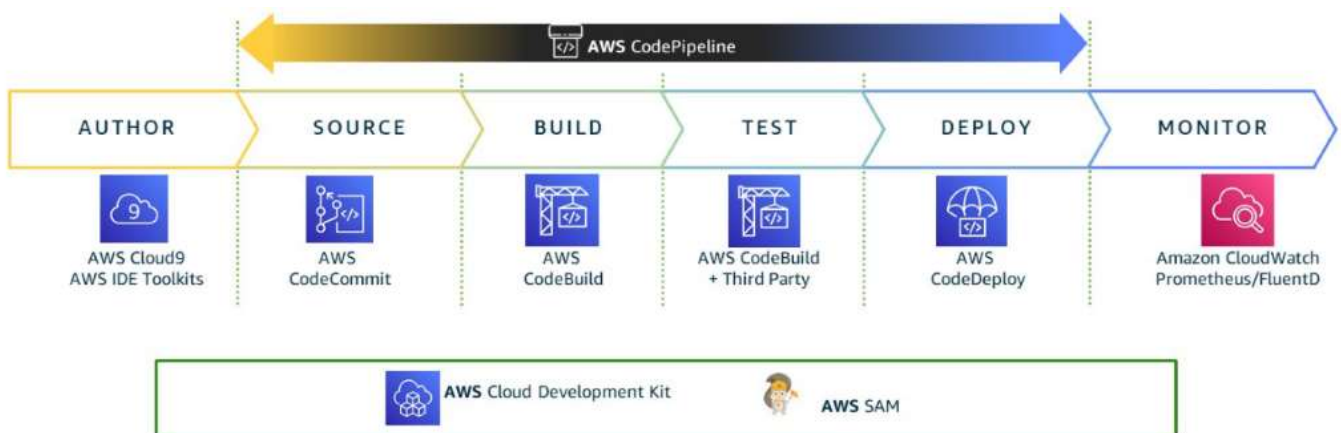
Formateur : Mohamed AIJOU

https://docs.aws.amazon.com/whitepapers/latest/cicd_for_5g_networks_on_aws/cicd-on-aws.html

<https://fr.wikipedia.org/wiki/CI/CD>

En **génie logiciel**, **CI/CD** (parfois écrit **CICD**) est la combinaison des pratiques d'**intégration continue** et de **livraison continue** ou de **déploiement continu**. Le CI/CD comble le fossé entre les activités et les équipes de développement et d'exploitation en imposant l'automatisation de la création, des tests et du déploiement des applications. Les pratiques **DevOps** modernes impliquent le développement continu, le test continu, l'intégration continue, le déploiement continu et la surveillance continue des applications logicielles tout au long de leur cycle de vie. La pratique CI/CD, ou pipeline CI/CD, constitue l'ossature des opérations DevOps modernes.

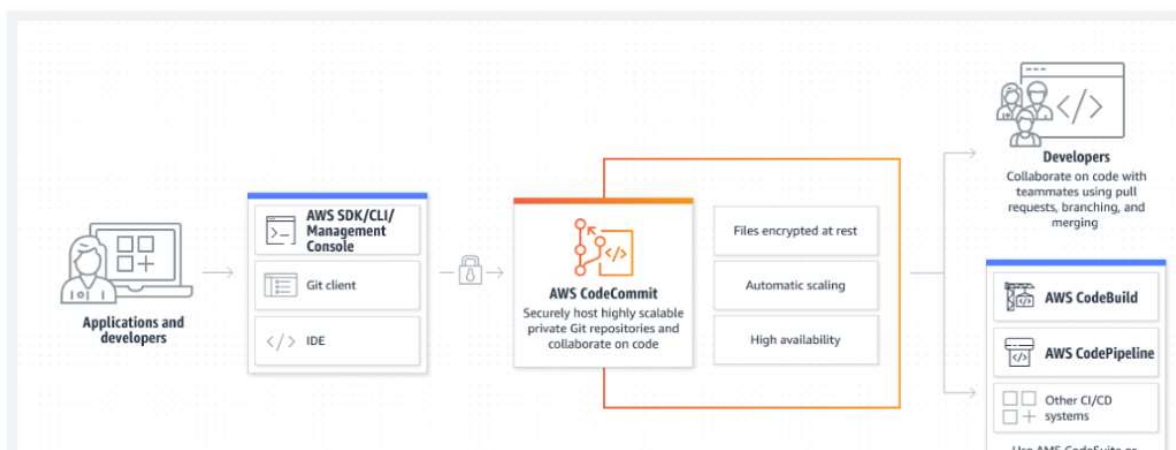
CI/CD can be pictured as a pipeline, where new code is submitted on one end, tested over a series of stages (source, build, test, staging, and production), and then published as production-ready code.



AWS Cloud9 : <https://aws.amazon.com/fr/cloud9/>

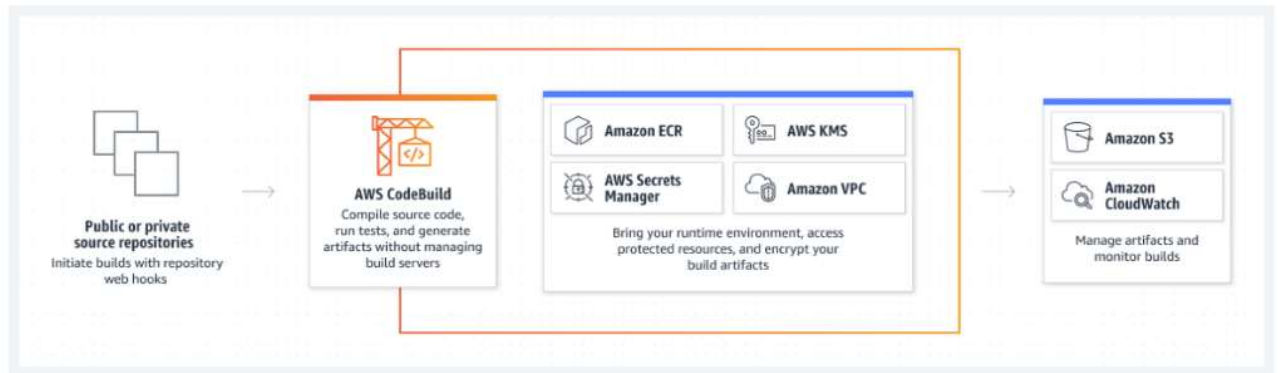
AWS Code Commit : <https://aws.amazon.com/fr/codecommit/>

AWS CodeCommit est un service de contrôle de sources géré, sécurisé et hautement évolutif qui héberge des référentiels Git privés.



AWS Code Build : <https://aws.amazon.com/fr/codebuild/>

AWS CodeBuild est un service d'intégration entièrement géré qui compile votre code source, exécute des tests et produit des packages logiciels prêts à être déployés.



AWS Code Deploy : <https://aws.amazon.com/fr/codedeploy/>

AWS CodeDeploy est un service de déploiement entièrement géré qui automatise le déploiement de logiciels vers divers services de calcul, tels qu'Amazon Elastic Compute Cloud (EC2), Amazon Elastic Container Service (ECS), AWS Lambda et vos serveurs sur site. Utilisez AWS CodeDeploy pour automatiser les déploiements de logiciels, éliminant ainsi la nécessité d'opérations manuelles sujettes aux erreurs.



Code pipeline : connecte les autres services entre eux

1ère étape : créer un référentiel sur **Code Commit**

Créer un référentiel

Créez un référentiel sécurisé pour stocker et partager votre code. Commencez par taper un nom et une description pour votre référentiel. Les noms de référentiel sont inclus dans les URL de ce référentiel.

Paramètres de référentiel

Nom du référentiel

100 caractères maximum. D'autres limites s'appliquent.

Créer - facultatif

1 000 caractères maximum

Balises

☐ Activer Amazon CodeGuru Reviewer pour Java et Python - facultatif

☐ Activer Amazon CodeGuru Reviewer pour Java et Python - *Facultatif*

Obtenez des recommandations pour améliorer la qualité du code Java et Python de toutes les demandes d'extraction dans ce référentiel.

Un rôle lié à un service sera créé dans IAM en votre nom s'il n'existe pas.

Annuler Créer

✓ **Succès**
Référentiel créé avec succès

Outils pour développeurs > CodeCommit > Référentiels > benoit-m2i-devops

benoit-m2i-devops

▼ Étapes de connexion

HTTPS SSH HTTPS (GRC)

Une fois le référentiel créé, il faut générer des informations d'identification git HTTPS sur notre compte dans la console IAM :

Informations d'identification Git HTTPS pour AWS CodeCommit (0)

Générez un nom d'utilisateur et un mot de passe que vous pouvez utiliser pour authentifier des connexions HTTPS sur des référentiels AWS CodeCommit. Vous pouvez disposer
fois. [Learn more](#)

Actions ▼ Générer des informations d'identification

Nom d'utilisateur	Création
Aucune information d'identification	
Générer des informations d'identification	

On récupère le fichier avec les éléments d'authentification et on le stock :

Générer des informations d'identification

✓ Vos nouvelles informations d'identification sont disponibles.

Enregistrez votre nom d'utilisateur et votre mot de passe ou téléchargez le fichier d'informations d'identification.

C'est la seule fois où vous pouvez voir le mot de passe ou le télécharger. Vous ne pourrez pas le récupérer plus tard. Cependant, vous pouvez réinitialiser votre mot de passe à tout moment.

Vous pouvez utiliser ces informations d'identification lorsque vous vous connectez à partir de votre ordinateur local ou à partir d'outils qui nécessitent un nom d'utilisateur et un mot de passe statiques. [En savoir plus](#)

Nom d'utilisateur
benoit-at-639962416620

Mot de passe
CDyNq8tu8ui7xHAWrQSeANMdVtmU44D+07qV/aV8eMo= Masquer

Télécharger les informations d'identification Fermer

On rentre dans notre dossier cloné : on créer un fichier et on push le fichier sur notre code commit en respectant les étapes :

```
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp
$ git clone https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/benoit-m2i-devops
Cloning into 'benoit-m2i-devops'...
warning: You appear to have cloned an empty repository.
```

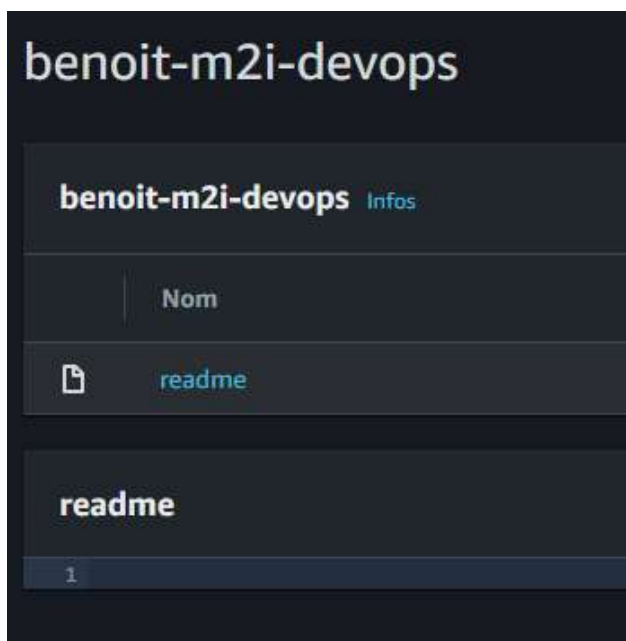
Git add .

Git commit -m "mon message"

Git push

```
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp
$ cd benoit-m2i-devops/
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ ls
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ touch readme
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git add .
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git commit -m "Paris est magique !"
[master (root-commit) 56c9366] Paris est magique !
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 readme
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 221 bytes | 221.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/benoit-m2i-devops
 * [new branch]      master -> master
```

On voit que notre fichier apparaît bien :



A partir de là : nous allons récupérer les fichier de notre projet, les rajouter dans notre dossier, et les push :

```
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git add .
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   app.js
    new file:   cron.yaml
    new file:   fichier1
    new file:   index.html
    new file:   package.json
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
```



```
Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git commit -m "ajout des fichiers du projet web"
[master a5e7fef] ajout des fichiers du projet web
5 files changed, 146 insertions(+)
create mode 100644 app.js
create mode 100644 cron.yaml
create mode 100644 fichier1
create mode 100644 index.html
create mode 100644 package.json

Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.95 KiB | 1.96 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/benoit-m2i-devops
56c9366..a5e7fef master -> master

Administrateur@LIL-JTJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ |
```

On peut vérifier nos branches dans le menu branche : ici nous avons uniquement la branche master :



On peut également accéder au menu notification pour créer des alertes (peut remonter par un service sns si configuré ou un chatbot slack)

Outils pour développeurs > CodeCommit > Référentiels > benoit-m2i-devops > Paramètres

Créer une règle de notification

Les règles de notification définissent un abonnement aux événements qui se produisent au niveau de vos ressources. Lorsque ces événements ont lieu, des notifications sont envoyées aux cibles que vous désignez. Vous pouvez gérer vos préférences de notification dans Paramètres. [Infos](#)

Paramètres de règles de notification

Nom de la notification

Type de détail
Sélectionnez le niveau de détail de votre choix pour les notifications. [En savoir plus sur les notifications et la sécurité](#)

☒ **Complet**
Comprend toutes les informations supplémentaires sur les événements fournies par la ressource ou la fonction de notifications.

☐ **Basique**
Ne comprend que les informations fournies dans les événements de la ressource.

Événements qui déclenchent des notifications

[Sélectionner aucun](#) [Sélectionner tout](#)

Comments	Approvals	Pull request	Branches and tags
<input type="checkbox"/> On commits	<input type="checkbox"/> Status changed	<input type="checkbox"/> Source updated	<input type="checkbox"/> Created
<input type="checkbox"/> On pull requests	<input type="checkbox"/> Rule override	<input type="checkbox"/> Created	<input type="checkbox"/> Deleted
		<input type="checkbox"/> Status changed	<input type="checkbox"/> Updated
		<input type="checkbox"/> Merged	

Création projet code build :

Configuration du projet

Nom du projet

demo-nodejs-benoit-m2i

Un nom de projet doit comporter entre 2 et 255 caractères. Il peut contenir les lettres A-Z et a-z, les chiffres 0-9 et les caractères spéciaux - et _.

Description - *facultatif*

Badge de génération - *facultatif*

☐ Activer le badge de génération

Activer la limite de génération simultanée - *facultatif*

Limitez le nombre de générations simultanées autorisées pour ce projet.

☐ Limiter le nombre de générations simultanées que ce projet peut démarrer

▼ Configuration supplémentaire

Balises

Clé	Valeur	
Name	demo-nodejs-benoit-m2i	Supprimer la balise

Ajouter une balise

Source Ajouter une source

Source 1 - Principale

Fournisseur de la source

AWS CodeCommit ▼

Référentiel

Q benoit-m2i-devops X

Type de référence

Sélectionnez le type de référence de version de la source qui contient votre code source.

☒ Branche

☐ Balise Git

☐ ID de validation

Branche

Sélectionnez une branche qui contient le code à générer.

master ▼

ID de validation - *facultatif*

Sélectionnez un ID de validation. Cela peut réduire la durée de votre génération.

Q

Version de la source [Infos](#)

refs/heads/master

a5e7fef6 ajout des fichiers du projet web

► Configuration supplémentaire

Profondeur du clone Git, Sous-modules Git

Environnement

Image d'environnement

☒ Image gérée
Utiliser une image gérée par AWS CodeBuild

☐ Image personnalisée
Spécifier une image Docker

Système d'exploitation

Amazon Linux 2 ▼

Les environnements d'exécution des langages de programmation sont maintenant inclus dans l'image standard d'Ubuntu 18.04, ce qui est recommandé pour les nouveaux projets CodeBuild créés dans la console. Consultez Images Docker fournies par CodeBuild pour plus d'informations [🔗](#).

Environnement(s) d'exécution

Standard ▼

Image

aws/codebuild/amazonlinux2-aarch64-standard:2.0 ▼

Version d'image

Toujours utiliser la dernière image pour cette version d'exécution ▼

Privilégié

☐ Activer cet indicateur si vous souhaitez créer des images Docker ou pour que vos générations bénéficient de privilèges élevés

Rôle de service

☒ **Nouveau rôle de service**
Créer un rôle de service dans votre compte

☐ **Rôle de service existant**
Choisir un rôle de service existant dans votre compte

Nom du rôle

codebuild-demo-nodejs-benoit-m2i-service-role

Taper le nom de votre rôle de service

Artefacts

Ajouter un artefact

Artefact 1 - Principal

Type

Amazon S3 ▼

Vous pouvez choisir Aucun artefact si vous exécutez des tests ou transmettez une image Docker à Amazon ECR.

Nom du compartiment

Q bucket-tp-sdk-benoit X

Nom

Nom du dossier ou du fichier compressé dans le compartiment qui contiendra vos artefacts de sortie. Utilisez Artifacts packaging (Emballage des artefacts) sous Additional configuration (Configuration supplémentaire) pour choisir si vous souhaitez utiliser un dossier ou un fichier compressé. Si le nom n'est pas indiqué, le nom du projet est utilisé par défaut.

☐ **Activer la gestion sémantique des versions**
Utiliser le nom d'artefact spécifié dans le fichier buildspec

Chemin - facultatif

Chemin du dossier ou du fichier ZIP de sortie de génération.

Par exemple : MyPath/MyArtifact.zip.

Type d'espace de noms - facultatif

Aucun(e) ▼

Choisissez ID de génération pour insérer l'ID de la génération dans le chemin vers le fichier zip ou le dossier de sortie de génération (par exemple, MyPath/MyBuildID/MyArtifact.zip). Sinon, choisissez Aucun.

Emballage des artefacts

☐ **Aucun(e)**
Les fichiers d'artefact seront chargés dans le compartiment.

☒ **Zip**
AWS CodeBuild charge les artefacts dans un fichier compressé qui est placé dans le compartiment spécifié.

Journaux

CloudWatch

☒ **Journaux CloudWatch - facultatif**
Cochez cette option pour charger les journaux de sortie de génération dans CloudWatch.

Nom du groupe

Nom du flux

S3

☐ **Journaux S3 - facultatif**
Cochez cette option pour charger les journaux de sortie de génération dans S3.

Annuler

Créer un projet de génération

✔ **Projet créé**

Vous avez créé le projet suivant avec succès : demo-nodejs-benoit-m2i

Outils pour développeurs

>

CodeBuild

>

Projets de génération

>

demo-nodejs-benoit-m2i

On peut maintenant démarrer la génération de notre projet :

demo-nodejs-benoit-m2i

Notifier

Partager

Modifier

Supprimer un projet de génération

Démarrer la génération avec des remplacements

Démarrer la génération

Configuration

Fournisseur de la source AWS CodeCommit	Référentiel principal benoit-m2i-devops	Emplacement de chargement des artefacts bucket-tp-sdk-benoit	Badge de génération Désactivé
Générations publiques Désactivé			

On va créer notre fichier buildspec.yaml :

benoit-m2i-devops

Notifier

master

Créer une demande d'extraction

URL du clone

benoit-m2i-devops

Infos

Ajouter un fichier

Créer un fichier

Charger le fichier

Nom

Outils pour développeurs

>

CodeCommit

>

Référentiels

>

benoit-m2i-devops

>

Fichier

Créer un fichier

benoit-m2i-devops

Infos

1 # !

Valider les modifications dans master

Nom de fichier

Par exemple, file.txt

buildspec.yaml

benoit-m2i-devops/buildspec.yaml

Nom de l'auteur

Benoit

Documentation utile pour la création d'un buildspec :

<https://docs.aws.amazon.com/codebuild/latest/userguide/build-spec-ref.html>

Code du fichier :

Créer un fichier

benoit-m2i-devops


```

benoit-mzi-devops  infos
1  version: 0.2
2  phases:
3    install:
4      runtime-versions:
5        nodejs: latest
6      commands:
7        - echo "installing something"
8    pre_build:
9      commands:
10       - echo "we are in the pre build phase"
11    build:
12      commands:
13       - echo "we are in the build block"
14       - echo "we will run some tests"
15       - echo "Congratulations" index.html
16    post_build:
17      commands:
18       - echo "we are in the post build phase"

```

Une fois le fichier complété : on valide

L'action effectuée sera équivalente à un **git commit/git push**

On peut voir les actions effectuées par le fichier dans la partie **validation** :

Validation 50b71d2a5368c699a99ba102b061ab85a67ada3b Copier

▼ Détails

Auteur	Authored date	Committer
Benoît test.test@testeurdebuild.buildcom	Maintenant	Benoît test.test@testeurdebuild.buildcom
Date de validation	Validation parent	
Maintenant	a5e7fe62716be8fb4330c91b05c465da58e9b9a	
Message de validation		
Added buildspec.yaml		

```

buildspec.yaml  Ajouté
1  + version: 0.2
2  + phases:
3  +   install:
4  +     runtime-versions:
5  +       nodejs: latest
6  +     commands:
7  +       - echo "installing something"
8  +   pre_build:
9  +     commands:
10 +       - echo "we are in the pre build phase"
11 +   build:
12 +     commands:
13 +       - echo "we are in the build block"
14 +       - echo "we will run some tests"
15 +       - echo "Congratulations" index.html
16 +   post_build:
17 +     commands:
18 +       - echo "we are in the post build phase"

```

Dans code build : on démarre la génération

La génération a démarré
 Vous avez démarré avec succès la génération suivante : demo-nodejs-benoit-m2i:22771e8a-0a9f-4fd6-9acd-0ea307e34143.

Outils pour développeurs > CodeBuild > Projets de génération > demo-nodejs-benoit-m2i > demo-nodejs-benoit-m2i:22771e

On peut voir les détails de la phase dans le sous-menu "détails de la phase" :

Nom	Statut
SUBMITTED	✔ Opération réussie
QUEUED	✔ Opération réussie
PROVISIONING	✔ Opération réussie
DOWNLOAD_SOURCE	✔ Opération réussie
INSTALL	✔ Opération réussie
PRE_BUILD	✔ Opération réussie
BUILD	✔ Opération réussie
POST_BUILD	✔ Opération réussie
UPLOAD_ARTIFACTS	✔ Opération réussie
FINALIZING	✔ Opération réussie
COMPLETED	✔ Opération réussie

Le détail du journal est disponible dans la partie **Journaux de génération** :

Journaux de génération | **Détail de la phase** | Rapports | Variables d'environnement | Détails de la génération

Affichage des 56 dernières lignes du journal de génération. [Afficher le journal complet](#)

^ Afficher les journaux précédents

1

[Container] 2023/02/23 15:54:29 Waiting for agent ping

2

[Container] 2023/02/23 15:54:30 Waiting for DOWNLOAD_SOURCE

3

[Container] 2023/02/23 15:54:34 Phase is DOWNLOAD_SOURCE

4

[Container] 2023/02/23 15:54:34 CODEBUILD_SRC_DIR=/codebuild/output/src945028934/src/git-codecommit.eu-west-1.amazonaws.com

5

[Container] 2023/02/23 15:54:34 YAML location is /codebuild/output/src945028934/src/git-codecommit.eu-west-1.amazonaws.com

6

[Container] 2023/02/23 15:54:34 Not setting HTTP client timeout for source type codecommit

7

[Container] 2023/02/23 15:54:34 Processing environment variables

8

[Container] 2023/02/23 15:54:34 Resolved 'nodejs' runtime alias 'latest' to '12'.

9

[Container] 2023/02/23 15:54:34 Selecting 'nodejs' runtime version '12' based on manual selections...

10

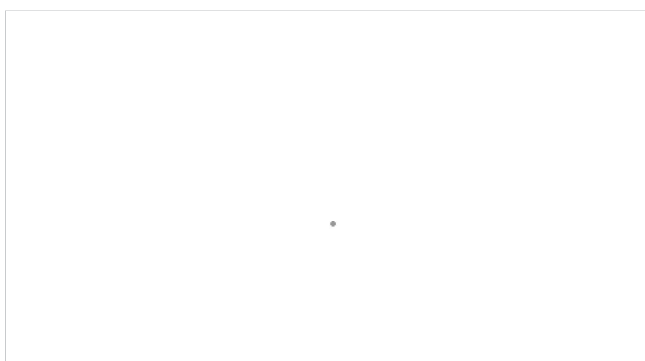
[Container] 2023/02/23 15:54:34 Running command echo "Installing Node.js version 12 ..."

11

Installing Node.js version 12 ...

12

Après l'étape du code build, nous devons déployer notre code avec **Code Deploy** : pour cela nous avons besoin d'une nouvelle machine :



Détails	Sécurité	Mise en réseau	Stockage	Vérifications de statut	Surveillance	Balises
<div> <div>▼ Résumé de l'instance Informations</div> <div> <div>ID d'instance</div> <div>i-0469347cb5ccf82e8 (node_benoit_application_test_m2i)</div> </div> <div> <div>Adresse IPv4 publique</div> <div>34.243.112.152 adresse ouverte</div> </div> <div> <div>Adresse IPv6</div> <div>–</div> </div> <div> <div>État de l'instance</div> <div>⌚ En attente</div> </div> <div> <div>Type de nom d'hôte</div> <div>Nom de l'adresse IP: ip-172-31-31-226.eu-west-1.compute.internal</div> </div> <div> <div>Nom DNS de l'IP privé (IPv4 uniquement)</div> <div>ip-172-31-31-226.eu-west-1.compute.internal</div> </div> <div> <div>Réponse à un nom DNS de ressource privée</div> <div>IPv4 (A)</div> </div> <div> <div>Type d'instance</div> <div>t2.micro</div> </div> </div>						

On va devoir installer un agent code deploy sur notre machine EC2 pour assurer la communication :

<https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-install-cli.html>

Nous allons utiliser ce lien pour installer l'agent sur notre machine :

<https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-install-linux.html>

```
Données utilisateur - optional Informations
Enter user data in the field.

#!/bin/bash

# Installing CodeDeploy Agent
sudo yum update
sudo yum install ruby

# Download the agent (replace the region)
wget https://aws-codedeploy-eu-west-3.s3.eu-west-3.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto
sudo service codedeploy-agent status
```

```
#!/bin/bash
# Installing CodeDeploy Agent
sudo yum update -y
sudo yum install ruby -y

# Download the agent (replace the region)
wget https://aws-codedeploy-eu-west-3.s3.eu-west-3.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto
sudo service codedeploy-agent status
```

```
Complete!
I, [2023-02-24T09:32:54.808997 #6434] INFO -- : Update check complete.
I, [2023-02-24T09:32:54.809095 #6434] INFO -- : Stopping updater.
[ec2-user@ip-172-31-16-93 ~]$ sudo service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 6519
```

Nous allons ensuite créer un rôle :

Étape 1
Sélectionner une entité de confiance

Étape 2
Ajouter des autorisations

Étape 3
Nommer, vérifier et créer

Sélectionner une entité de confiance [Infos](#)

Type d'entité approuvée

☒ Service AWS

Autorisez les services AWS tels qu'EC2, Lambda ou autre à effectuer des actions dans ce compte.

☐ Compte AWS

Autorisez les entités d'autres comptes AWS qui appartiennent à vous à un tiers à effectuer des actions dans ce compte.

☐ Identité Web

Permet aux utilisateurs fédérés par le fournisseur d'identité web externe spécifié d'assumer ce rôle pour effectuer des actions dans ce compte.

☐ Fédération SAML 2.0

Autoriser les utilisateurs fédérés avec SAML 2.0 à partir d'un répertoire d'entreprise à effectuer des actions dans ce compte.

☐ Stratégie d'approbation personnalisée

Créez une stratégie d'approbation personnalisée pour permettre à d'autres utilisateurs d'effectuer des actions dans ce compte.

Cas d'utilisation

Autorisez un service AWS comme EC2, Lambda ou autres à effectuer des actions dans ce compte.

Cas d'utilisation courants

☒ EC2

Allows EC2 instances to call AWS services on your behalf.

☐ Lambda

Allows Lambda functions to call AWS services on your behalf.



Politiques des autorisations (Sélectionnée(s) 1/889) [Infos](#)

Choisissez une ou plusieurs stratégies à attacher à votre nouveau rôle.

Filtrer les stratégies par nom de propriété ou de stratégie et appuyer sur Entrée.

"s3" ✕

Effacer les filtres

<input type="checkbox"/>	Nom de la politique ✎	Type ▼
<input type="checkbox"/>	+ KinesisFirehoseServicePolicy-KDS-S3-pAV05-eu-west-3	Gérées ...
<input type="checkbox"/>	+ KinesisFirehoseServicePolicy-PUT-S3-jew5D-eu-west-3	Gérées ...
<input type="checkbox"/>	+ KinesisFirehoseServicePolicy-PUT-S3-uowQI-eu-west-3	Gérées ...
<input type="checkbox"/>	+ s3	Gérées ...
<input type="checkbox"/>	+ s3-sdk	Gérées ...
<input type="checkbox"/>	+ S3_mfA	Gérées ...
<input type="checkbox"/>	+ s3_read_write	Gérées ...
<input type="checkbox"/>	+ strategie_s3	Gérées ...
<input type="checkbox"/>	+  AmazonDMSRedshiftS3Role	Gérées ...
<input checked="" type="checkbox"/>	+  AmazonS3FullAccess	Gérées ...

Informations du rôle

Nom du rôle

Saisissez un nom explicite pour identifier ce rôle.

role_s3_ec2_benoit_m2i

64 caractères maximum. Utilisez des caractères alphanumériques, ainsi que les caractères « +=, @- _ ».

Description

Ajoutez une brève explication de ce rôle.

Allows EC2 instances to call AWS services on your behalf.

1 000 caractères maximum. Utilisez des caractères alphanumériques, ainsi que les caractères '+ = , @ - _ '.

Étape 1 : sélectionner des entités de confiance

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "sts:AssumeRole"  
8       ],  
9       "Principal": {  
10        "Service": [  
11          "ec2.amazonaws.com"  
12        ]  
13      }  
14    }  
15  ]  
16 }
```

Second role pour code deploy :

Ajouter des autorisations [Infos](#)

Politiques des autorisations (Sélectionnée(s) 1/889) [Infos](#)

Choisissez une ou plusieurs stratégies à attacher à votre nouveau rôle.

🔍 Filtrer les stratégies par nom de propriété ou de stratégie et appuyer su

"codedeploy" ✕

Effacer les filtres

☐ Nom de la politique ↗

☑️ AmazonEC2RoleforAWSCodeDeploy

<input type="checkbox"/>	Nom du rôle	Entités de confiance
<input type="checkbox"/>	benoit-role-code-deploy-pipeline-m2i	Service AWS: ec2
<input type="checkbox"/>	codebuild-demo-nodejs-benoit-m2i-service-role	Service AWS: codebuild
<input type="checkbox"/>	role_s3_ec2_benoit_m2i	Service AWS: ec2

Type d'entité approuvée

☒ Service AWS

Autorisez les services AWS tels qu'EC2, Lambda ou autre à effectuer des actions dans ce compte.

☐ Compte AWS

Autorisez les entités d'autres comptes AWS qui appartiennent à vous à un tiers à effectuer des actions dans ce compte.

☐ Identité Web

Permet aux utilisateurs fédérés par le fournisseur d'identité web externe spécifié d'assumer ce rôle pour effectuer des actions dans ce compte.

- ☐ Fédération SAML 2.0
Autoriser les utilisateurs fédérés avec SAML 2.0 à partir d'un répertoire d'entreprise à effectuer des actions dans ce compte.
- ☐ Stratégie d'approbation personnalisée
Créez une stratégie d'approbation personnalisée pour permettre à d'autres utilisateurs d'effectuer des actions dans ce compte.

Cas d'utilisation

Autorisez un service AWS comme EC2, Lambda ou autres à effectuer des actions dans ce compte.

Cas d'utilisation courants

- ☐ EC2
Allows EC2 instances to call AWS services on your behalf.
- ☐ Lambda
Allows Lambda functions to call AWS services on your behalf.

Cas d'utilisation pour d'autres services AWS :

- CodeDeploy ▼
- ☒ CodeDeploy
Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.
- ☐ CodeDeploy for Lambda
Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.
- ☐ CodeDeploy - ECS
Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.

Liste des rôles à avoir pour notre pipeline :

IAM > Rôles

Rôles (120) Infos

Un rôle IAM est une identité que vous pouvez créer et qui dispose d'autorisations spécifiques avec des informations d'identification valides pendant de courtes durées. Les rôles peuvent être endossés par des entités de confiance.

Q benoit X 3 correspondances

<input type="checkbox"/>	Nom du rôle	Entités de confiance
<input type="checkbox"/>	benoit-role-code-deploy-pipeline-m2i	Service AWS: ec2
<input type="checkbox"/>	role-code-deploy-pipeline-benoit-m2i	Service AWS: codedeploy
<input type="checkbox"/>	role_s3_ec2_benoit_m2i	Service AWS: ec2

Outils pour développeurs

AWS CodeDeploy

Automatisez les déploiements de code pour maintenir la disponibilité des applications

AWS CodeDeploy est un service de déploiement entièrement géré qui automatise les déploiements de logiciels pour des services de calcul comme Amazon EC2, AWS Lambda, ainsi que vos serveurs sur site. AWS CodeDeploy vous permet de lancer rapidement et facilement de nouvelles fonctionnalités et d'éviter les temps d'arrêt pendant le déploiement d'une application, tout en gérant la complexité de la mise à jour de vos applications.

Créer un déploiement AWS CodeDeploy

Démarrez avec AWS CodeDeploy en créant votre première application de déploiement.

[Créer une application](#)

On va commencer par créer l'application : pour ce TP, nous allons choisir EC2 comme plateforme de calcul

Outils pour développeurs > CodeDeploy > Applications > Créer une application

Créer une application

Configuration de l'application

Nom de l'application

Entrer un nom d'application

100 caractères maximum

100 caractères maximum

Plateforme de calcul

Choisir une plateforme de calcul

EC2/Sur site ▼

Annuler Créer une application

✓ **Application créée**

Pour créer un déploiement, vous devez d'abord créer un groupe de déploiement.

Création du déploiement :

Créer un groupe de déploiement

Application

Application
benoit-application-test

Type de calcul
EC2/Sur site

Nom du groupe de déploiement

Entrer un nom de groupe de déploiement

benoit-application-test

100 caractères maximum

Rôle de service

Entrer un rôle de service

Saisissez un rôle de service doté d'autorisations CodeDeploy pour accorder à AWS CodeDeploy l'accès à vos instances cibles.

arn:aws:iam::639962416620:role/code-deploy-pipeline-benoit-m2i X

Type de déploiement

Choisir comment déployer votre application

☒ **Sur place**

Met à jour les instances dans le groupe de déploiement avec les révisions d'application les plus récentes. Lors d'un déploiement, chaque instance sera mise hors ligne brièvement pendant sa mise à jour.

☐ **Bleu/vert**

Remplace les instances dans le groupe de déploiement par de nouvelles instances et déploie la révision d'application la plus récente dans ces instances. Une fois les instances dans l'environnement de remplacement enregistrées avec un équilibreur de charge, l'enregistrement des instances de l'environnement d'origine est annulé et celles-ci peuvent être mises hors service.

On sélectionne instance EC2 + dans les tag, on utilise le nom de notre machine :

Configuration de l'environnement

Sélectionner une combinaison de groupes Amazon EC2 Auto Scaling, d'instances Amazon EC2 et d'instances sur site à ajouter à ce déploiement

☐ Groupes Auto Scaling Amazon EC2

☒ **Instances Amazon EC2**

Instance correspondante unique 1. [Cliquez ici pour en savoir plus](#)

Vous pouvez ajouter jusqu'à 3 groupes de balises pour des instances EC2 à ce groupe de déploiement.
Un groupe de balises : Le déploiement aura lieu dans toute instance identifiée par le groupe de balises.
Plusieurs groupes de balises : Le déploiement aura lieu uniquement dans les instances identifiées par tous les groupes de balises.

Groupe de balises 1

Clé

Valeur - *facultatif*

🔍 Name ✕

🔍 node-benoit-main-pipeline ✕

Supprimer la
balise

Ajouter une balise

+ Ajouter un groupe de balises

☐ Instances sur site

Correspondance d'instances

Instance correspondante unique 1. [Cliquez ici pour en savoir plus](#)

Configuration de l'agent avec AWS Systems Manager [Infos](#)

AWS Systems Manager installe l'agent CodeDeploy sur toutes les instances et le met à jour en fonction de la fréquence configurée.



Nous vous recommandons de configurer l'installation et les mises à jour de votre agent CodeDeploy avec AWS Systems Manager.

AWS Systems Manager offre plus de contrôle sur les mises à jour et restaurations de version de l'agent CodeDeploy que sur l'installation à l'aide d'autres méthodes. [En savoir plus](#)

Installation de l'agent AWS CodeDeploy

- ☒ Jamais
- ☐ Une seule fois
- ☐ Maintenant et planifier des mises à jour

Configuration de l'agent avec AWS Systems Manager [Infos](#)

AWS Systems Manager installe l'agent CodeDeploy sur toutes les instances et le met à jour en fonction de la fréquence configurée.



Nous vous recommandons de configurer l'installation et les mises à jour de votre agent CodeDeploy avec AWS Systems Manager.

AWS Systems Manager offre plus de contrôle sur les mises à jour et restaurations de version de l'agent CodeDeploy que sur l'installation à l'aide d'autres méthodes. [En savoir plus](#)

Installation de l'agent AWS CodeDeploy

- ☒ Jamais
- ☐ Une seule fois
- ☐ Maintenant et planifier des mises à jour



Groupe de déploiement créé

Outils pour développeurs > CodeDeploy > Applications > benoit-application-test > benoit-application-test

benoit-application-test

Détails du groupe de déploiement

Nom du groupe de déploiement	Nom de l'application
benoit-application-test	benoit-application-test
Type de déploiement	ARN de rôle de service
Sur place	arn:aws:iam::639962416620:role/role-code-deploy-pipeline-benoit-m2i
Restauration activée	Planificateur de mise à jour de l'agent
faux	Apprenez à planifier une mise à jour dans AWS Systems Manager

On retourne dans code commit pour créer le fichier **appspec.yaml** :

Documentation fichier appspec :

<https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file.html>

Exemple fichier appspec :

Here is an example of a correctly spaced AppSpec file:

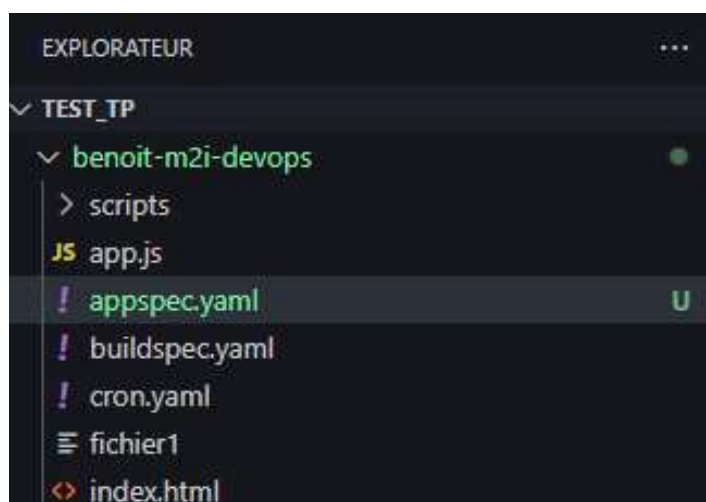
```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
    - location: scripts/create_test_db.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```



Pour créer ce fichier : on va retourner dans notre dépôt sur vscode, faire un **git pull** et créer un dossier **script** et un fichier **appspec**

```
Administrateur@LIL-JT3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git pull
remote: Counting objects: 3, done.
Unpacking objects: 100% (3/3), 596 bytes | 37.00 KiB/s, done.
From https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/benoit-m2i-devops
   a5e7fef..50b71d2  master    -> origin/master
Updating a5e7fef..50b71d2
Fast-forward
 buildspec.yaml | 18 ++++++
 1 file changed, 18 insertions(+)
 create mode 100644 buildspec.yaml

Administrateur@LIL-JT3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ ls
app.js  buildspec.yaml  cron.yaml  fichier1  index.html  package.json  readme

Administrateur@LIL-JT3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$
```



 package.json
 readme

On complète notre fichier :

```
! appspec.yaml U X
benoit-m2i-devops > ! appspec.yaml > {} hooks > [ ] ApplicationStop > {} 0 > runas
1  version: 0.0
2  os: linux
3  files:
4    - source: /index.html
5      destination: /var/www/html/
6  hooks:
7    BeforeInstall:
8      - location: scripts/install_dependencies
9        timeout: 300
10       runas: root
11     - location: scripts/start_server
12       timeout: 300
13       runas: root
14    ApplicationStop:
15      - location: scripts/stop_server
16        timeout: 300
17       runas: root
```

```
version: 0.0
os: linux
files:
- source: /index.html
  destination: /var/www/html/
hooks:
  BeforeInstall:
  - location: scripts/install_dependencies
    timeout: 300
    runas: root
  - location: scripts/start_server
    timeout: 300
    runas: root
  ApplicationStop:
  - location: scripts/stop_server
    timeout: 300
    runas: root
```

On va également créer nos fichiers dans le dossier scripts car ils sont appelé dans notre fichier **appspec.yaml** :

Fichier **install_dependencies** :

```
benoit-m2i-devops > scripts > $ install_dependencies
1  #!/bin/bash
2  yum install -y httpd
```

Fichier **start_server** :

```
benoit-m2i-devops > scripts > $ start_server
1  #!/bin/bash
2  service httpd start
```

Fichier **stop_server** :


```

benoit-m2i-devops > scripts > $ stop_server
1  #!/bin/bash
2  isExistApp = `pgrep httpd`
3  if [[ -n $isExistApp ]]; then
4  |   service httpd stop
5  fi

```

```

#!/bin/bash
isExistApp = `pgrep httpd`
if [[ -n $isExistApp ]]; then
service httpd stop
fi

```

On push nos modifications sur le dépôt AWS Commit :

```

Administrateur@LIL-3TJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git add .
warning: in the working copy of 'scripts/install_dependencies', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/start_server', LF will be replaced by CRLF the next time Git touches it

Administrateur@LIL-3TJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

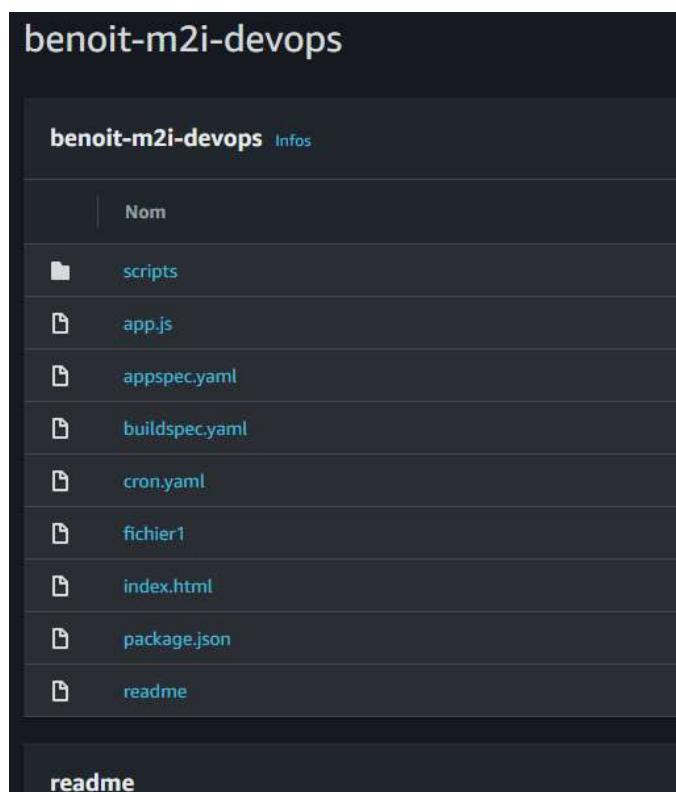
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   appspec.yaml
    modified:   readme
    new file:   scripts/install_dependencies
    new file:   scripts/start_server
    new file:   scripts/stop_server

Administrateur@LIL-3TJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git commit -m "add scripts + appspec.yaml"
[master 5369df8] add scripts + appspec.yaml
5 files changed, 27 insertions(+)
create mode 100644 appspec.yaml
create mode 100644 scripts/install_dependencies
create mode 100644 scripts/start_server
create mode 100644 scripts/stop_server

Administrateur@LIL-3TJ3KN3 MINGW64 ~/Documents/Infrastructure as Code/ci_cd/test_tp/benoit-m2i-devops (master)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 878 bytes | 878.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/benoit-m2i-devops
50b71d2..5369df8  master -> master

```

Vérification du push des fichiers :



1 Ce projet est un TP réalisé en formation avec Mr Mohamed AIJJOU

On modifie le rôle IAM de notre instance pour lui ajouter le rôle S3 que nous avons créé :

EC2 > Instances > i-0bd546efbd7743f37 > Modifier le rôle IAM

Modifier le rôle IAM Informations

Attachez un rôle IAM à votre instance.

ID d'instance
i-0bd546efbd7743f37 (node-benoit-main-pipeline)

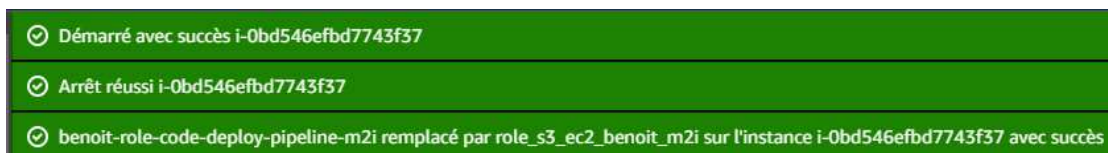
Rôle IAM
Sélectionner un rôle IAM à attacher à votre instance ou créer un rôle si vous n'en avez pas encore créé. Le rôle que vous sélectionnez remplace tous les rôles actuellement attachés à votre instance.

role_s3_ec2_benoit_m2i

Créer un nouveau rôle IAM

Annuler Mettre à jour le rôle IAM

Pour que la machine prenne en compte ce changement de rôle : on stop l'instance et on la redémarre :



Ensuite, dans code deploy : on créer un groupe de déploiement :

Déploiements				
Groupes de déploiement				
Révisions				
Groupes de déploiement				
Afficher les détails				
Modifier				
Créer un groupe de déploiement				
Q				
< 1 > ⚙				
Nom	Statut	Dernière tentative de déploiement	Dernier déploiement réussi	Nombre de déclencheurs
benoit-application-test	-	-	-	0

Outils pour développeurs > CodeDeploy > Applications > benoit-application-test > Créer un déploiement

Create deployment

Paramètres de déploiement

Application
benoit-application-test

Groupe de déploiement
Q benoit-application-test X

Plateforme de calcul
EC2/Sur site

Type de déploiement
Sur place

Type de révision
☒ Mon application est stockée dans Amazon S3 ☐ Mon application est stockée dans GitHub

Emplacement de la révision
Copier et coller le compartiment Amazon S3 dans lequel votre révision est stockée
Q
Aucune révision précédente

