

# Lab 3: Implémentation de CICD dans AWS avec AWS CodeCommit, CodeDeploy et CodePipeline.

## Scénario:

Une entreprise spécialisée dans la vente en ligne souhaite automatiser le déploiement de son application web pour améliorer l'efficacité et réduire les erreurs humaines. Actuellement, le processus de déploiement est manuel, ce qui entraîne des délais, des erreurs fréquentes, et un manque de visibilité sur l'état des versions en production. Pour résoudre ces problèmes, l'entreprise décide de faire appel à un AWS Cloud Engineer pour concevoir et mettre en place un pipeline CI/CD complet sur AWS.

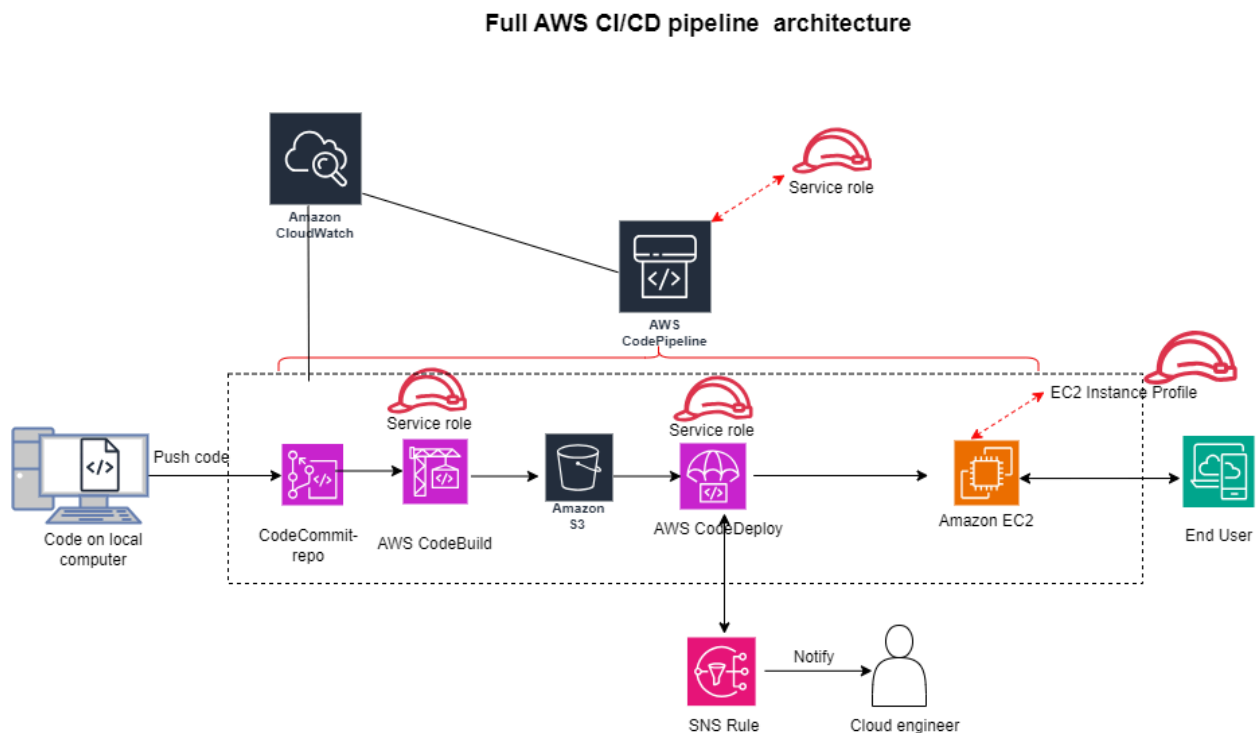
## Objectifs:

- Automatiser l'intégration continue (CI) pour s'assurer que chaque modification du code est automatiquement testée et validée.
- Automatiser le déploiement continu (CD) pour permettre un déploiement fluide et sans interruption de l'application sur l'environnement de production.
- Garantir la sécurité, la scalabilité et la résilience du pipeline CI/CD.

Coûts: free tier

Temps estimé de réalisation: 45 minutes

Architecture de solution:



## Réalisation

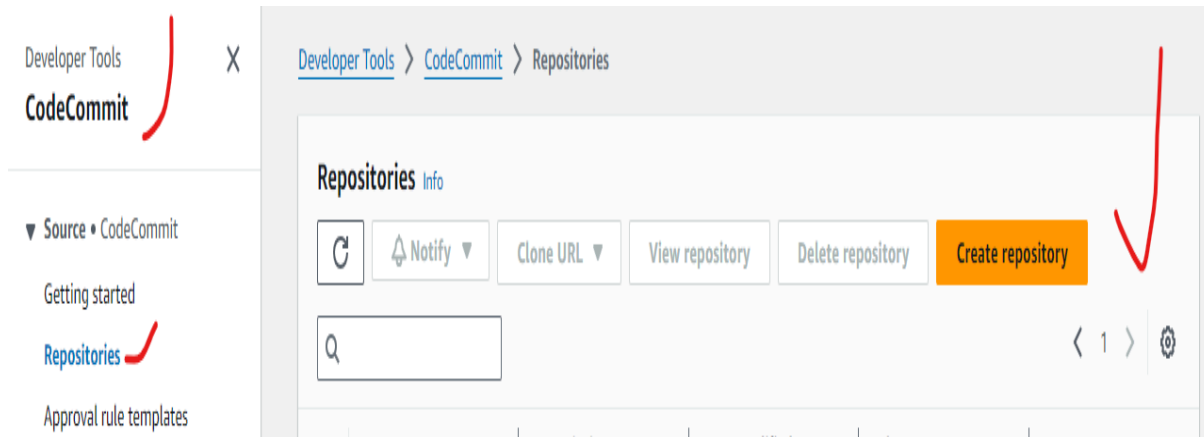
**Étape 1: Télécharger les informations d'identification HTTPS Git pour AWS CodeCommit à l'aide de la console IAM.**

1) Cliquez sur **users** dans le menu de gauche et cliquez sur votre utilisateur puis sur **security credentials** Faites défiler vers le bas jusqu'à **HTTPS Git credentials for AWS CodeCommit** et cliquez sur **Generate Credential**.

The screenshot shows the AWS IAM console interface. The left sidebar contains the following navigation items: Identity and Access Management (IAM), Search IAM, Dashboard, Access management (expanded), User groups, Users (highlighted with a red checkmark), Roles, Policies, Identity providers, Account settings, Access reports (expanded), and Access Analyzer. The main content area is divided into two sections. The top section, 'SSH public keys', shows 'No SSH public keys' and a button 'Upload SSH public key' with a red checkmark. The bottom section, 'HTTPS Git credentials for AWS CodeCommit (0)', shows 'No credentials' and a button 'Generate credentials' with a red checkmark. A red arrow points to the 'Generate credentials' button in the top right corner of the main content area.

## Étape 2 : Créer un dépôt CodeCommit

1) Ouvrez la console Amazon Codecommit et choisissez de créer un dépôt.



2) Sur la page Créer un référentiel, dans Nom du référentiel, saisissez un nom pour votre référentiel (par exemple, EAZYrepo). Sélectionnez ensuite Créer.

A screenshot of the 'Create repository' form in AWS CodeCommit. The breadcrumb trail is 'Developer Tools > CodeCommit > Repositories > Create repository'. The title 'Create repository' is at the top. Below it, a paragraph explains: 'Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.' The 'Repository settings' section contains a 'Repository name' field with 'EAZYRepo' entered, which is circled in red. Below the field is the text '100 characters maximum. Other limits apply.' There is also a 'Description - optional' text area and a 'Tags' section at the bottom.

Add tag

Additional configuration

AWS KMS key

Enable Amazon CodeGuru Reviewer for Java and Python - optional

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

Cancel

Create

3) Sur l'écran suivant, cliquez sur Copy à côté de Step3 : Clone the Repository (Nous l'utiliserons pour cloner ce dépôt vierge dans l'étape suivante).

Developer Tools

CodeCommit

Source • CodeCommit

Getting started

Repositories

Code

Pull requests

Commits

Branches

Git tags

Settings

Approval rule templates

Artifacts • CodeArtifact

Build • CodeBuild

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. [Learn how to create and configure an IAM user for accessing AWS CodeCommit.](#) | [Learn how to add team members to an AWS CodeStar Project.](#)

Step 2: Git credentials

Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. [Generate Git Credentials](#)

Step 3: Clone the repository

Clone your repository to your local computer and start working on code. Run the following command:

git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/EAZYRepo

Copy

Additional details

You can find more detailed instructions in the documentation. [View documentation](#)

EAZYRepo

Info

Add file

### Etape:3 Ajouter un exemple de code à votre dépôt CodeCommit

Project proposed by Lahda Biassou Alphonsine

git clone <GIT Clone Address> puis entrez le nom d'utilisateur et le mot de passe des utilisateurs IAM HTTPS Git Credentials que nous avons déjà téléchargés. Toutes ces étapes sont décrites en capture.

```

C:\> Invite de commandes
Microsoft Windows [version 10.0.19045.4046]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\LADHA ALPHONCINE>cd C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting>git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/EAZYRepo
Cloning into 'EAZYRepo'...
warning: auto-detection of host provider took too long (>2000ms)
warning: see https://aka.ms/gcm/autodetect for more information.
warning: auto-detection of host provider took too long (>2000ms)
warning: see https://aka.ms/gcm/autodetect for more information.
warning: You appear to have cloned an empty repository.

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting>

```

[illegible]

```
C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting>cd EAZYrepo

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    css/
    deleted.html
    error.html
    fonts/
    img/
    index.html
    js/

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>
```

```
C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>git add -A
warning: in the working copy of 'css/bootstrap.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'css/site.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'deleted.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'error.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'fonts/glyphicons-halflings-regular.svg', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'js/bootstrap.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'js/jquery-1.11.2.min.js', LF will be replaced by CRLF the next time Git touches it

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>
```

```
C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   css/bootstrap.css
    new file:   css/site.css
    new file:   deleted.html
    new file:   error.html
    new file:   fonts/glyphicons-halflings-regular.eot
    new file:   fonts/glyphicons-halflings-regular.svg
    new file:   fonts/glyphicons-halflings-regular.ttf
    new file:   fonts/glyphicons-halflings-regular.woff
    new file:   fonts/glyphicons-halflings-regular.woff2
    new file:   img/bg-large.png
    new file:   img/bg-small.png
    new file:   img/email.png
    new file:   img/form-email.png
    new file:   img/form-password.png
    new file:   img/logo.png
    new file:   index.html
    new file:   js/bootstrap.js
    new file:   js/jquery-1.11.2.min.js

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>
```

```
C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>git commit -m "premier commit"
[master (root-commit) de17647] premier commit
18 files changed, 9825 insertions(+)
create mode 100644 css/bootstrap.css
create mode 100644 css/site.css
create mode 100644 deleted.html
create mode 100644 error.html
create mode 100644 fonts/glyphicons-halflings-regular.eot
create mode 100644 fonts/glyphicons-halflings-regular.svg
create mode 100644 fonts/glyphicons-halflings-regular.ttf
create mode 100644 fonts/glyphicons-halflings-regular.woff
create mode 100644 fonts/glyphicons-halflings-regular.woff2
create mode 100644 img/bg-large.png
create mode 100644 img/bg-small.png
create mode 100644 img/email.png
create mode 100644 img/form-email.png
create mode 100644 img/form-password.png
create mode 100644 img/logo.png
create mode 100644 index.html
create mode 100644 js/bootstrap.js
create mode 100644 js/jquery-1.11.2.min.js

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>
```

```
C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>git push
warning: auto-detection of host provider took too long (>2000ms)
warning: see https://aka.ms/gcm/autodetect for more information.
warning: auto-detection of host provider took too long (>2000ms)
warning: see https://aka.ms/gcm/autodetect for more information.
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 4 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (24/24), 942.20 KiB | 10.59 MiB/s, done.
Total 24 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/EAZYRepo
 * [new branch]      master -> master

C:\Users\LADHA ALPHONCINE\Downloads\WebsiteHosting\EAZYRepo>
```

Avant de créer le serveur EC2, créez un **rôle d'instance** qui permet l'installation et la gestion de le CodeDeploy agent sur l'instance. Le **CodeDeploy agent** est un progiciel qui permet à une instance d'être utilisée dans CodeDeploy des déploiements. Vous attachez également des politiques qui permettent à l'instance de récupérer les fichiers que le CodeDeploy agent utilisé pour déployer votre application et de permettre à l'instance d'être gérée par SSM.

#### Pour créer un rôle d'instance:

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>).
2. Dans le tableau de bord de la console, choisissez Rôles.



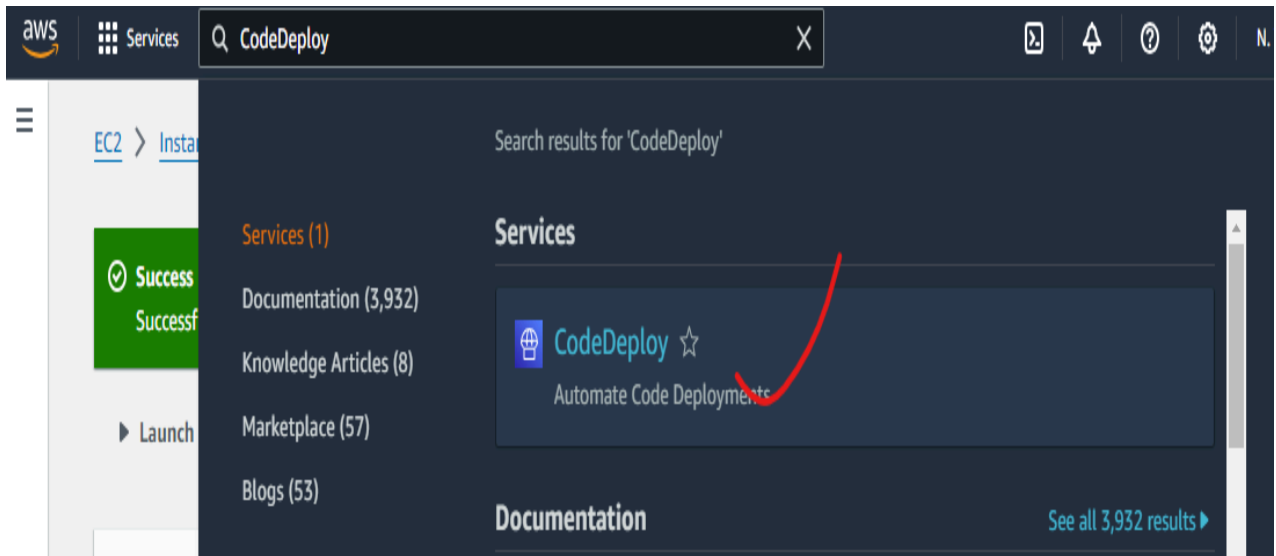
3. Sélectionnez Créer un rôle.
4. Sous Sélectionner le type d'entité de confiance, **sélectionnez Service AWS**. Sous Choisir un cas d'utilisation, **sélectionnez EC2**. Sous Select your use case (Sélectionner votre cas d'utilisation), choisissez EC2. Sélectionnez Next: Permissions (Étape suivante : autorisations).
5. Recherchez et sélectionnez la politique nommée AmazonEC2RoleforAWSCodeDeploy.
6. Recherchez et sélectionnez la politique nommée AmazonSSMManagedInstanceCore. Choisissez Suivant : Balises.
7. Choisissez Suivant : Vérification. Saisissez un nom pour le rôle (par exemple, EC2InstanceRole). Sélectionnez Créer un rôle.

#### **Étape 4: création d'un serveur de déploiement EC2 avec un AMI nommé Amazon Linux**

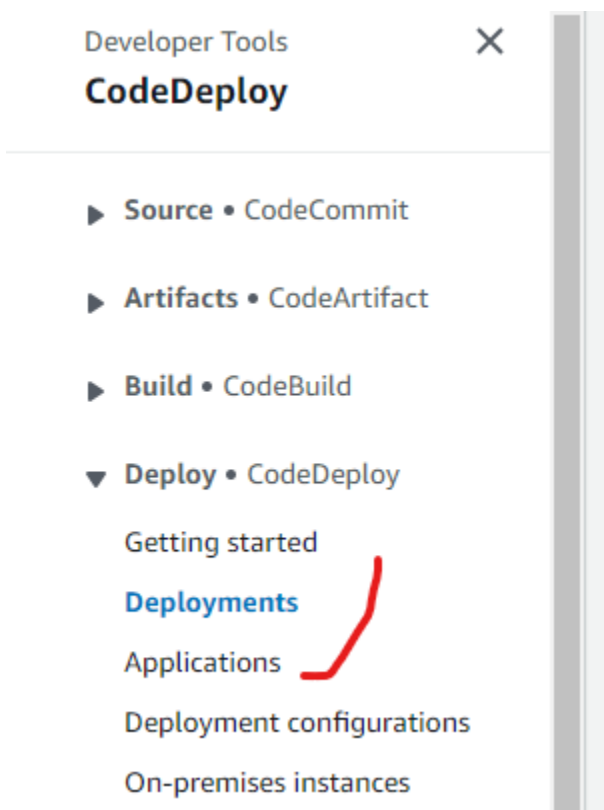
1. Ouvrez la console Amazon EC2
2. Dans la navigation latérale, choisissez Instances, puis sélectionnez Launch instances en haut de la page.
3. Pour Name (Nom), entrez **MyCodePipelineDemo**. Cela affecte à l'instance une balise Key of Name et une balise Value de MyCodePipelineDemo. Vous créerez ultérieurement une CodeDeploy application qui déploiera l'exemple d'application sur cette instance. CodeDeploy sélectionne les instances à déployer en fonction des balises.
4. Sous Images de l'application et du système d'exploitation (Amazon Machine Image), recherchez l'option AMI Amazon Linux avec le AWS logo et assurez-vous qu'elle est sélectionnée. (Cette AMI est décrite comme l'AMI Amazon Linux 2 (HVM) et est étiquetée « éligible au niveau gratuit ».)
5. Sous Type d'instance, choisissez le t2.micro type éligible au niveau gratuit comme configuration matérielle de votre instance.

6. Sous Paire de clés (connexion), choisissez une paire de clés ou créez-en une.  
Vous pouvez également choisir Proceed sans paire de clés.
7. Sous Paramètres réseau, procédez comme suit.  
Dans Attribuer automatiquement une adresse IP publique, assurez-vous que le statut est Activé.
  - a. Groupe de sécurité, choisissez Créer un groupe de sécurité.
  - b. Dans la ligne correspondant à SSH, sous **Type de source, sélectionnez Mon adresse IP.**
  - c. Choisissez Ajouter un groupe de sécurité, **choisissez HTTP, puis sous Type de source, sélectionnez Mon adresse IP.**
8. Développez **Advanced Details (Détails avancés)**. Dans le profil d'instance IAM, choisissez le rôle IAM que vous avez créé lors de la procédure précédente (par exemple, **EC2InstanceRole**).
9. Sous Résumé, sous Nombre d'instances, entrez 1.
10. Choisissez Launch instance (Lancer une instance).

## **Etape 5 : création d'un déploiement dans CodeDeploy**



- Création d'une application



aws

Services

Search

[Alt+S]

N. Virginia

lahda @ 4019-1528-2046

Developer Tools

CodeDeploy

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

▼ Deploy • CodeDeploy

Getting started

Deployments

Developer Tools > CodeDeploy > Applications

Applications

Notify ▾

View details

Deploy application

Create application

Q

< 1 > ⚙

Application name	Compute platform	Created
No results		
There are no results to display.		

# Create application

## Application configuration

### Application name

Enter an application name

100 character limit

### Compute platform

Choose a compute platform

### Tags

- Création de déploiement

✓ Application created

Create a notification rule for this application

X

In order to create a new deployment, you must first create a deployment group.

[Developer tools](#) / [CodeDeploy](#) / [Applications](#) / [AppDeploy](#)

AppDeploy

🔔 Notify ▼

Delete application

Application details

Name	Compute platform
AppDeploy	EC2/On-premises

Deployments

**Deployment groups**

Revisions

Deployment groups

View details

Edit

Create deployment group

< 1 > ⚙

Name	Status	Last attempted de...	Last successful de...	Trigger count
------	--------	----------------------	-----------------------	---------------

- **Creation de rôle IAM pour CodeDeploy**

## Identity and Access Management (IAM)



Search IAM

### Dashboard

#### ▼ Access management

User groups

Users

Roles



[IAM](#) > Roles

### Roles (8) [Info](#)



Delete

Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

< 1 >



Role name



Trusted entities

Last activity



[aws-elasticbeanstalk-service-role](#)

AWS Service: elasticbeanstalk

-

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

## Select trusted entity [Info](#)

### Trusted entity type

☒ AWS service

Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ Web identity

Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.



## Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

CodeDeploy

Choose a use case for the specified service.

Use case

☒ 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.

Cancel

Next

aws

Services

Search

[Alt+S]

Global

lahda @ 4019-1528-20

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

## Add permissions [Info](#)

Permissions policies (1) [Info](#)

The type of role that you selected requires the following policy.

Policy name	Type
<a href="#">AWSCodeDeployRole</a>	AWS managed

► Set permissions boundary - optional

Cancel

Previous

Next

- Spécification de nom du rôle et création

The screenshot shows the AWS IAM console interface for creating a new role. The top navigation bar includes the AWS logo, a 'Services' menu, a search bar, and user information. The left sidebar shows the navigation path: IAM > Roles > Create role. The main content area is titled 'Name, review, and create' and is divided into two sections: 'Role details' and 'Permissions'. The 'Role details' section contains two input fields: 'Role name' and 'Description'. Both fields have the text 'CodeDeployServiceRole' entered. Red handwritten lines are drawn over the input fields. The 'Permissions' section is currently empty.

aws Services Search [Alt+S] Global lahda

IAM > Roles > Create role

Step 1  
[Select trusted entity](#)

Step 2  
[Add permissions](#)

Step 3  
**Name, review, and create**

### Name, review, and create

#### Role details

**Role name**  
Enter a meaningful name to identify this role.

CodeDeployServiceRole

Maximum 64 characters. Use alphanumeric and '+=, @, \_' characters.

**Description**  
Add a short explanation for this role.

CodeDeployServiceRole

Maximum 1000 characters. Use alphanumeric and '+=, @, \_' characters.

Permissions policy summary

Policy name	Type	Attached as
<a href="#">AWSCodeDeployRole</a>	AWS managed	Permissions policy

### Step 3: Add tags

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create role

- **Création de groupe de déploiement**

## Deployment group name

Enter a deployment group name

DeploymentGroup

100 character limit

## Service role

Enter a service role

Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

Project proposed by Lahda Biassou Alphonsine

## Deployment type

Choose how to deploy your application

☒ In-place

Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

☐ Blue/green

Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

## Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

☐ Amazon EC2 Auto Scaling groups

☒ Amazon EC2 instances

0 unique matched instances. [Click here for details](#)

## Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

☐ Amazon EC2 Auto Scaling groups

☒ Amazon EC2 instances

1 unique matched instance. [Click here for details](#)

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key

Q Name

X

Value - optional

Q ServerCICD

X

Remove tag

Add tag

+ Add tag group

## Agent configuration with AWS Systems Manager [Info](#)



**Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent.**  
Make sure the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. [Learn more](#)

### Install AWS CodeDeploy Agent

- ☐ Never
- ☐ Only once
- ☒ Now and schedule updates

Basic scheduler

Cron expression

14

Days

## Deployment settings

### Deployment configuration

Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

CodeDeployDefault.AllAtOnce

or

Create deployment configuration

### Load balancer

Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.

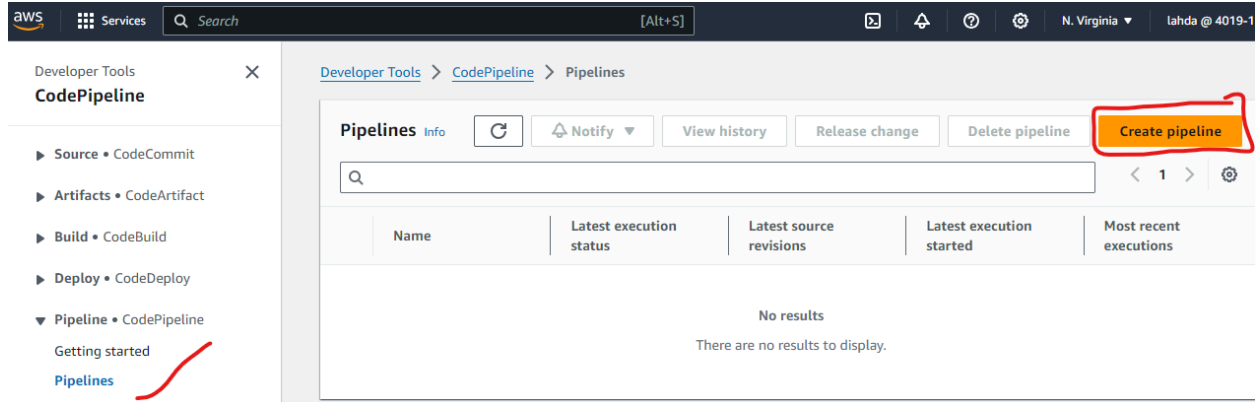
☐ Enable load balancing

► Advanced - optional

Cancel

Create deployment group

## Étape 6: création et configuration du pipeline avec CodePipeline



**Pipeline name**  
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

**Pipeline type**  
The pipeline type determines the pipeline structure and availability of parameters such as triggers. Pipeline type selection will impact features and pricing. [Which pipeline is right for me?](#)

☐ V1 ☒ V2

**Execution mode**  
Choose the execution mode for your pipeline. This determines how the pipeline is run.

☐ Superseded  
A more recent execution can overtake an older one. This is the default.

☒ Queued (Pipeline type V2 required)  
Executions are processed one by one in the order that they are queued.

☐ Parallel (Pipeline type V2 required)  
Executions don't wait for other runs to complete before starting or finishing.

**Service role**

☒ New service role  
Create a service role in your account

☐ Existing service role  
Choose an existing service role from your account

### Service role



#### New service role

Create a service role in your account



#### Existing service role

Choose an existing service role from your account

### Role name

AWSCodePipelineServiceRole-us-east-1-PipelineCICD

Type your service role name



Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

### Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

Add variable

You can add up to 50 variables.

### Advanced settings

Cancel

Next

Step 1

[Choose pipeline settings](#)

Step 2

**Add source stage**

Step 3

Add build stage

Step 4

Add deploy stage

Step 5

Review

## Add source stage Info

Step 2 of 5

### Source

#### Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit

#### Repository name

Choose a repository that you have already created where you have pushed your source code.

EAZYRepo

#### Branch name

Choose a branch of the repository

master

#### Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ Amazon CloudWatch Events (recommended)

☐ AWS CodePipeline

**Branch name**  
Choose a branch of the repository

Q master X

**Change detection options**  
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ **Amazon CloudWatch Events (recommended)**  
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

☐ **AWS CodePipeline**  
Use AWS CodePipeline to check periodically for changes

**Output artifact format**  
Choose the output artifact format.

☒ **CodePipeline default**  
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

☐ **Full clone**  
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

Cancel Previous **Next**

**Add deploy stage**

Step 5  
Review

**Deploy**

**Deploy provider**  
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

**Region**

US East (N. Virginia)

**Application name**  
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

Q AppDeploy X

**Deployment group**  
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

Q DeploymentGroup X

Cancel Previous **Next**



Step 1

Choose pipeline settings

Step 2

Add source stage

Step 3

Add build stage

Step 4

Add deploy stage

Step 5

Review

Review

Info

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name

PipelineCICD

Pipeline type

V2

Execution mode

QUEUED

Artifact location

A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name

AWSCodePipelineServiceRole-us-east-1-PipelineCICD

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS CodeDeploy

ApplicationName

AppDeploy

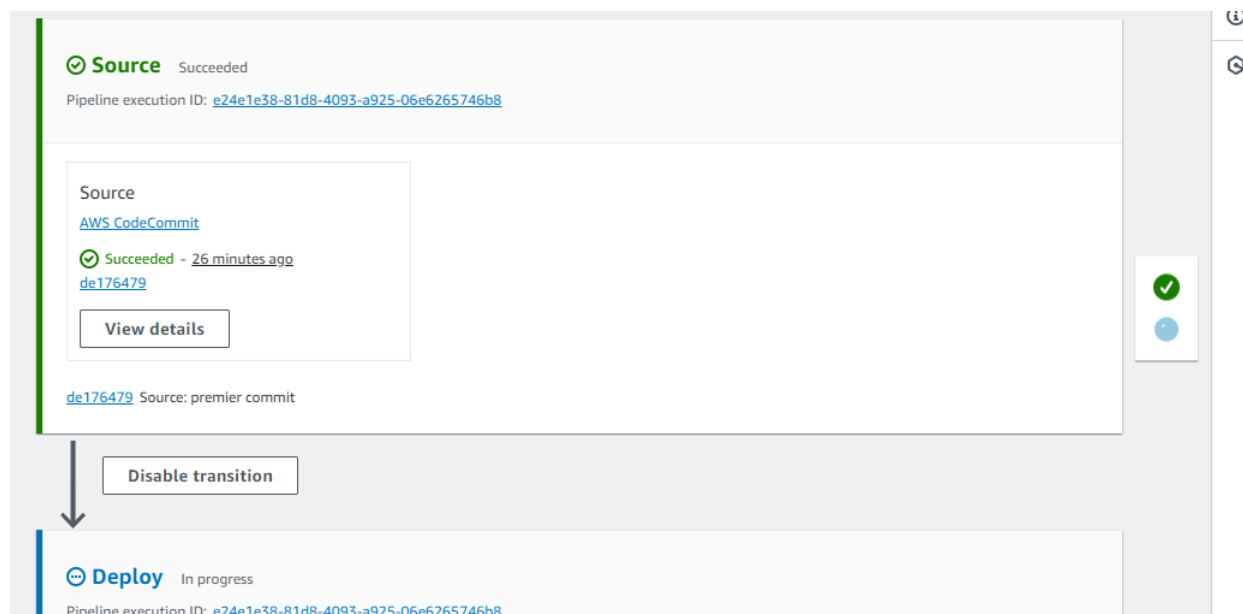
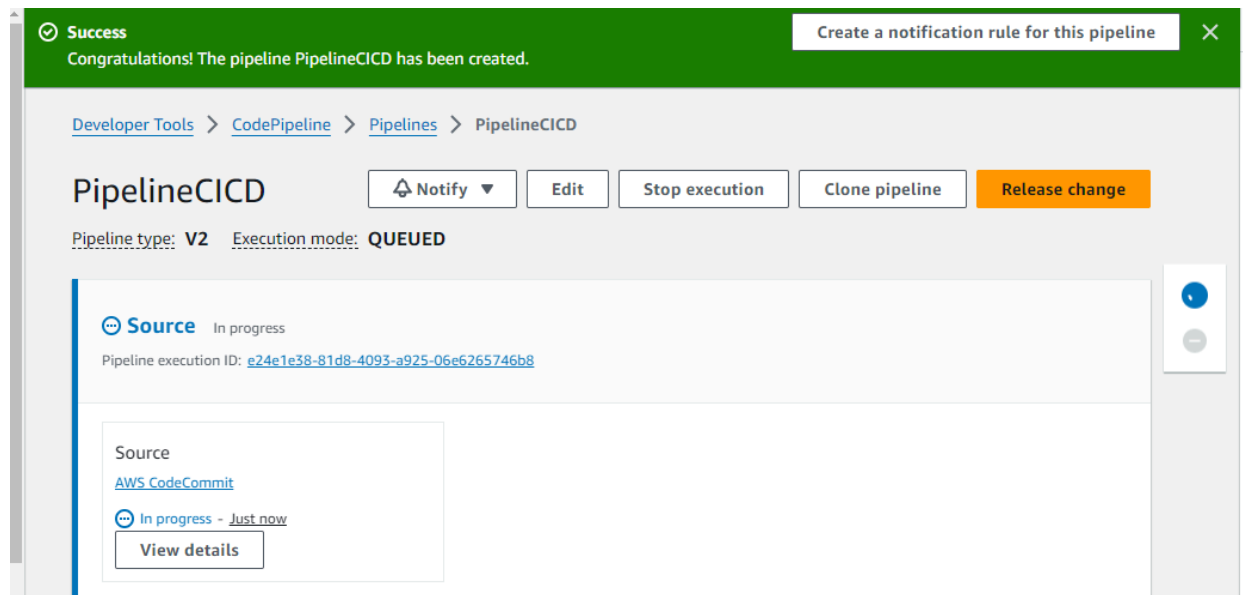
DeploymentGroupName

DeploymentGroup

Cancel

Previous

Create pipeline



Bravo, vous venez d'implémenter un CICD dans AWS. Et n'oubliez pas de nettoyer votre environnement en supprimant les ressources créées.