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.

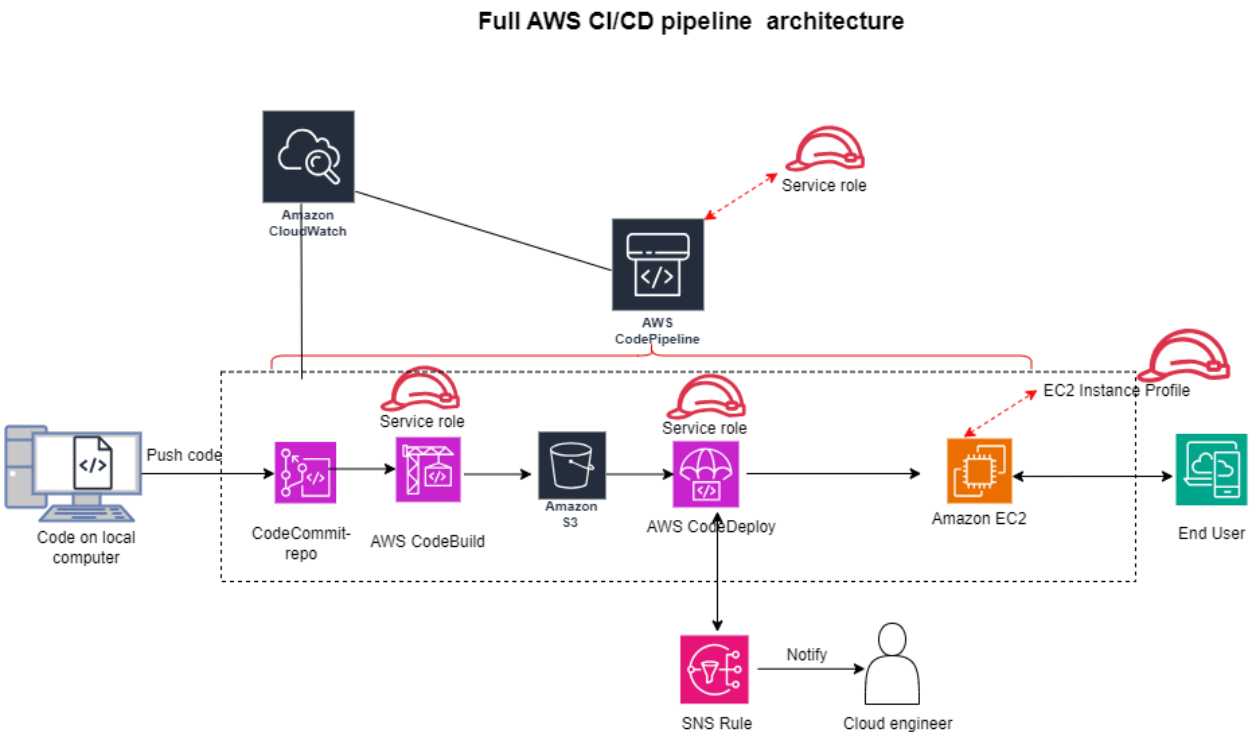
Objectifs:

- 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

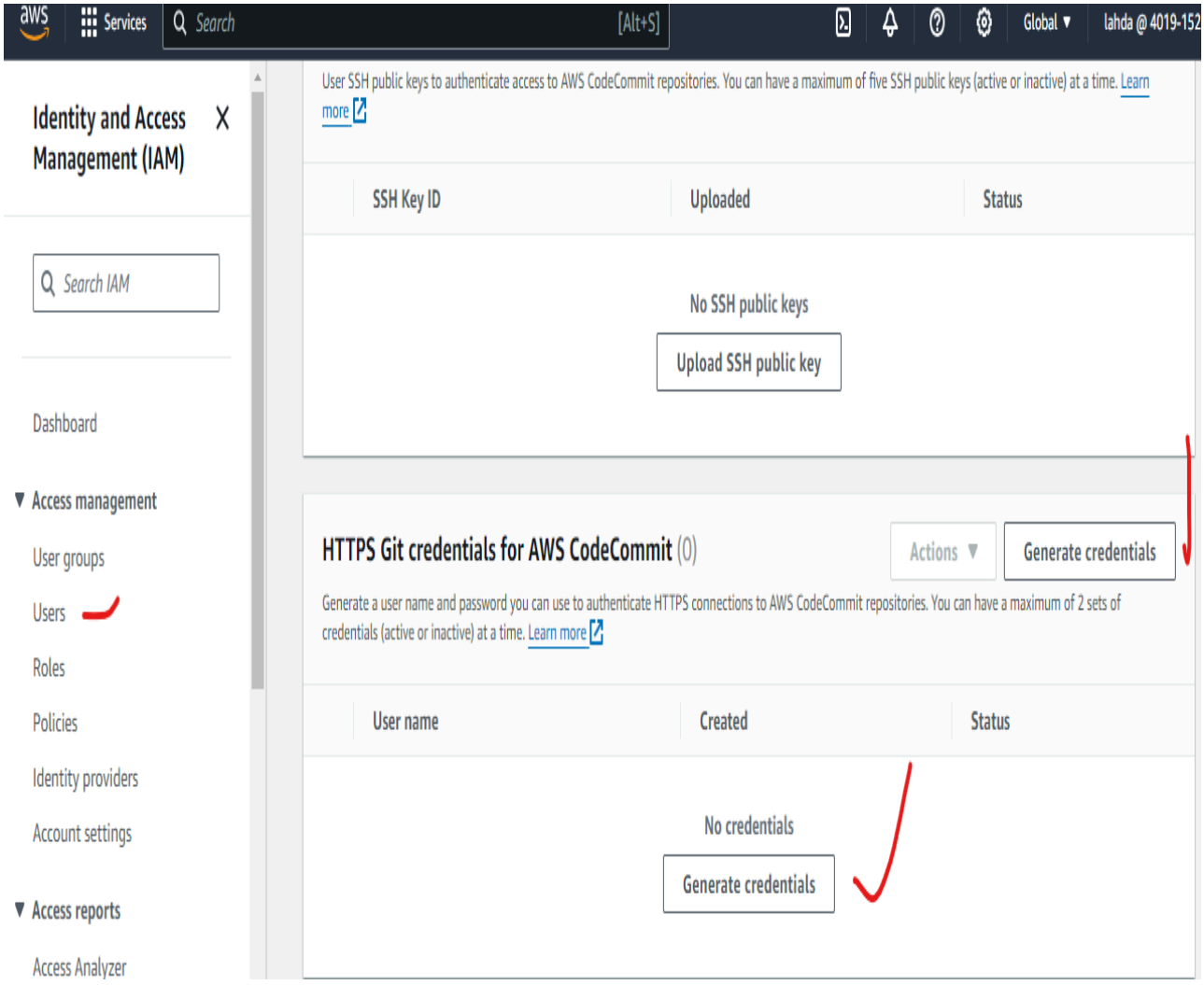
Architecture de solution:



Réalisation

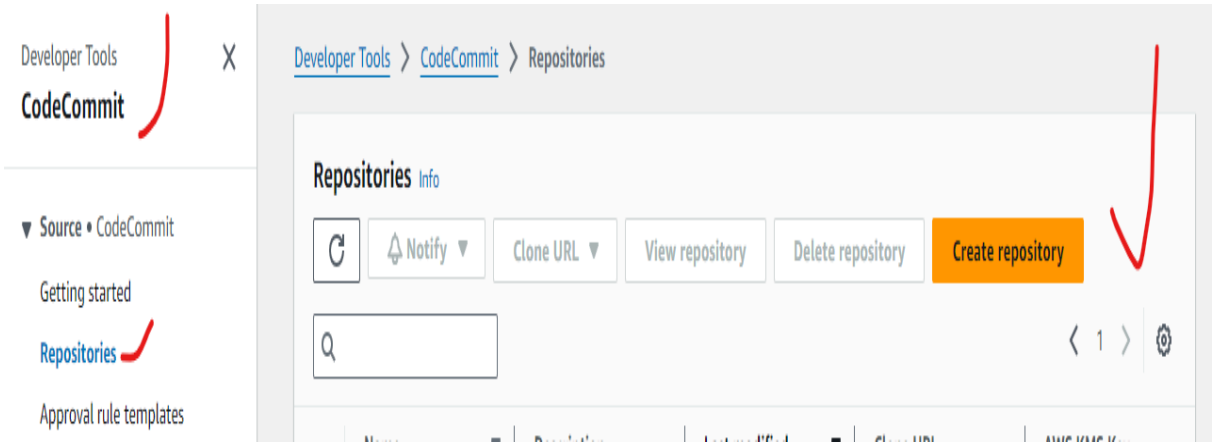
Étape 1: Télécharger les informations d'identification HTTPS Git pour AWS CodeCommit à l'aide de la console IAM.

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.**

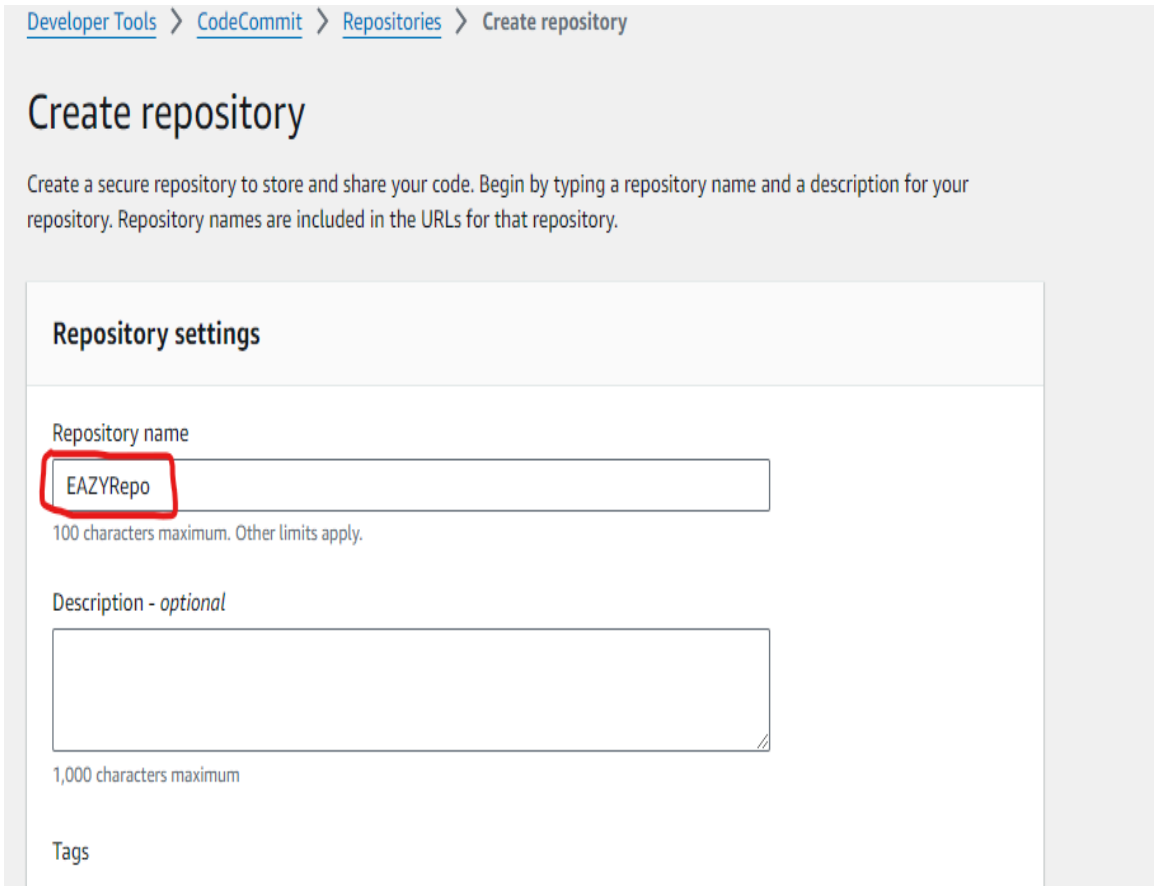


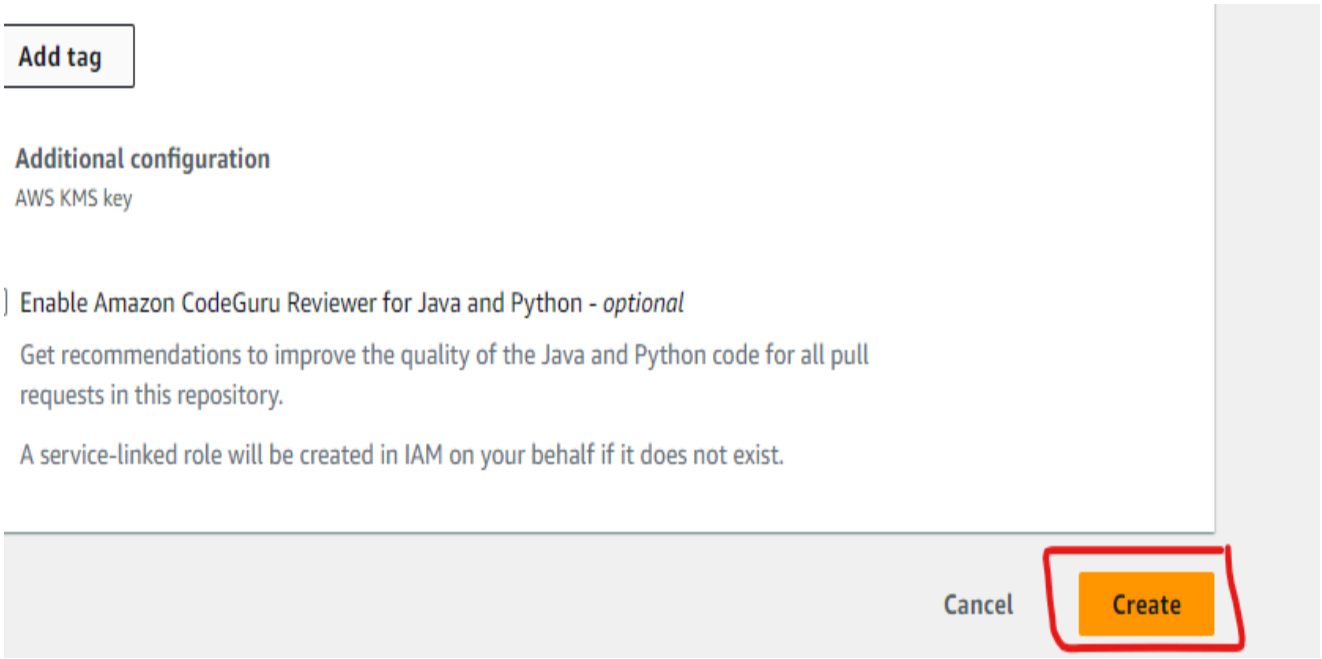
Étape 2 : Créer un dépôt CodeCommit

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

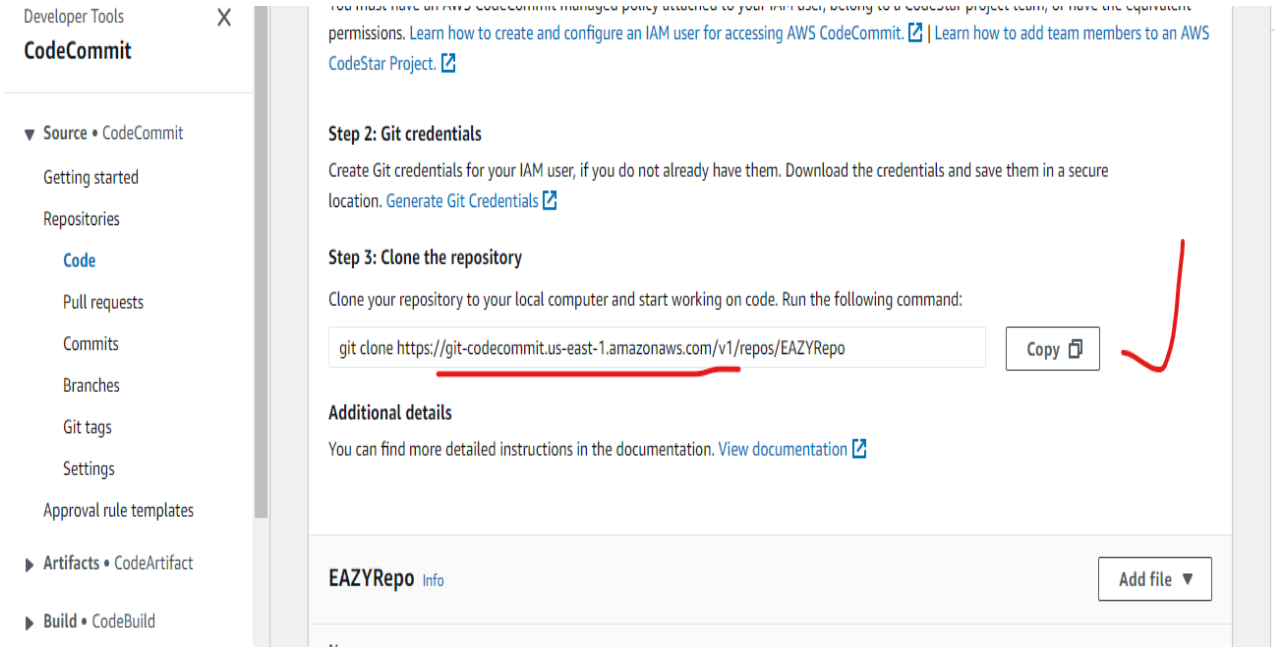


2) Sur la page Créer un référentiel, dans Nom du référentiel, saisissez un nom pour votre référentiel (par exemple, EAZYrepo). Sélectionnez ensuite Créer.





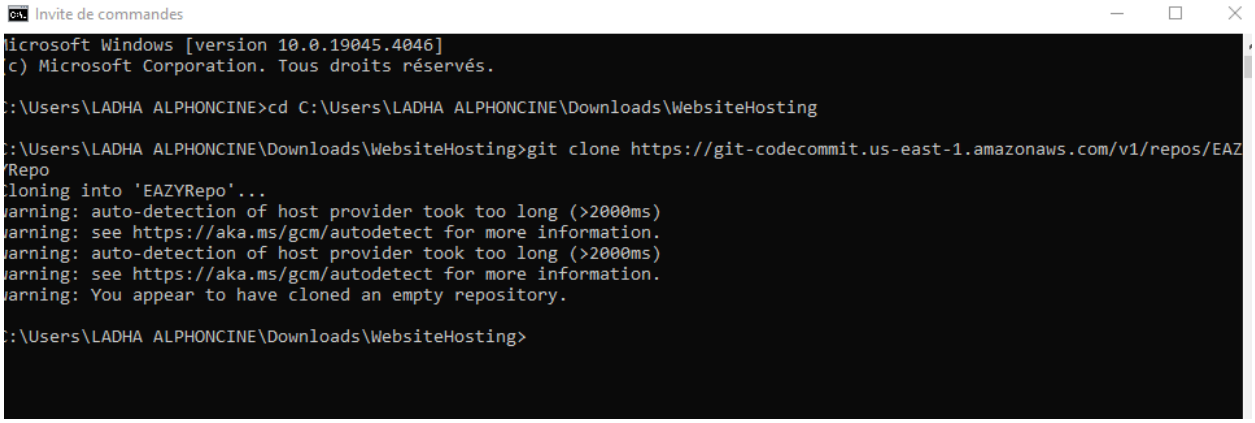
3) Sur l'écran suivant, cliquez sur Copy à côté de Step3 : Clone the Repository (Nous l'utiliserons pour cloner ce dépôt vierge dans l'étape suivante).

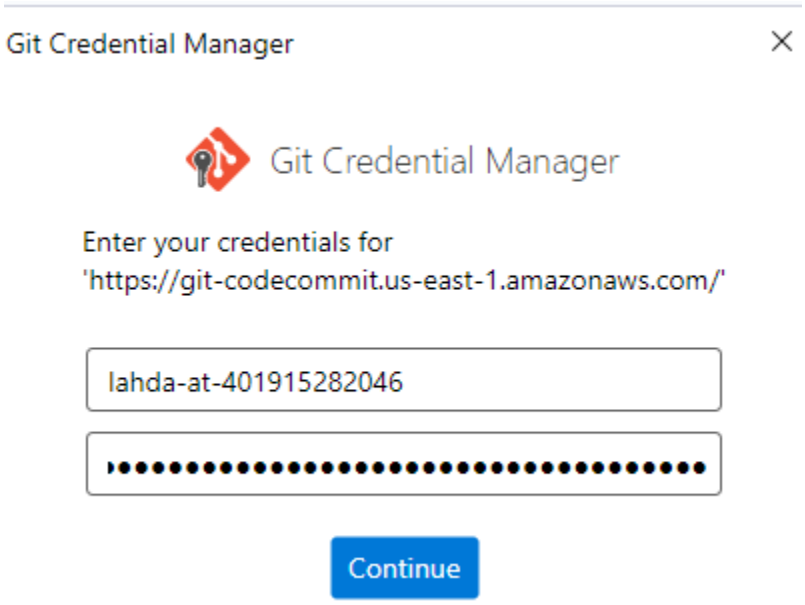


Etape:3 Ajouter un exemple de code à votre dépôt CodeCommit

1) Exécutez la commande suivante pour cloner le dépôt, en remplaçant l'adresse de clonage GIT par celle que vous avez copiée dans l'étape précédente.

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:\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>
```

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

:\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 age**nt 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.

Etape 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.

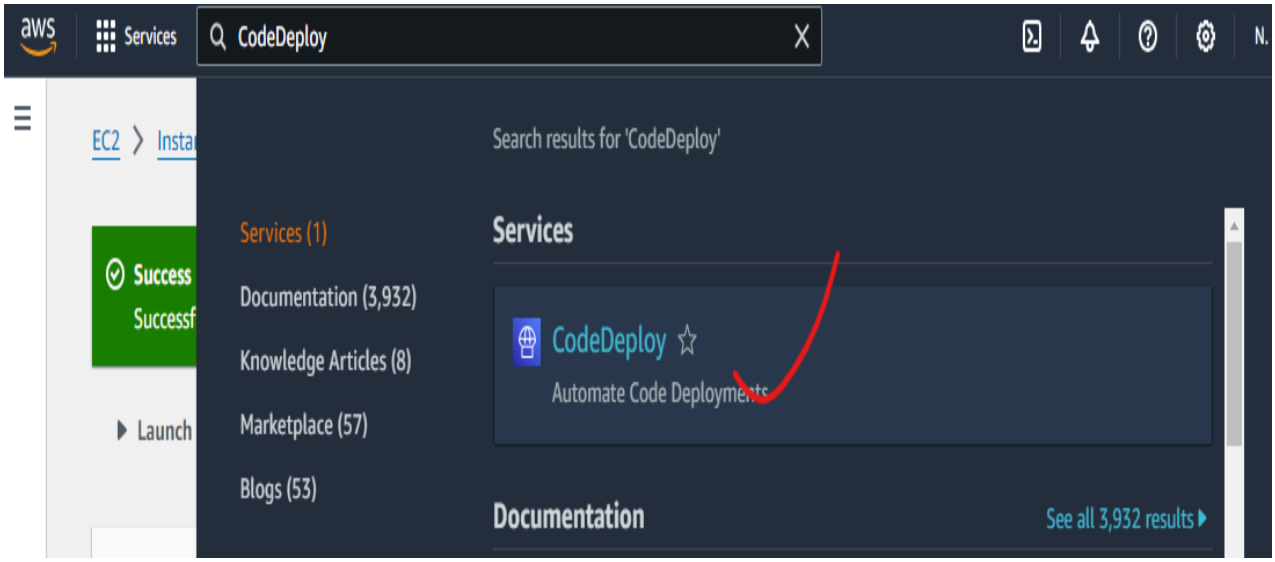
6. Sous Paire de clés (connexion), choisissez une paire de clés ou créez-en une.
   Vous pouvez également choisir Proceed sans paire de clés.

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**).

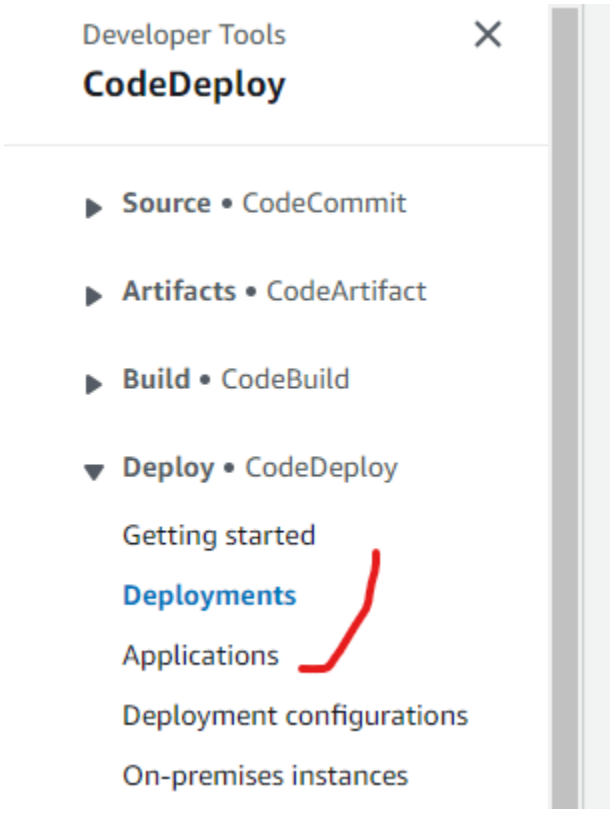
9. Sous Résumé, sous Nombre d'instances, entrez 1.

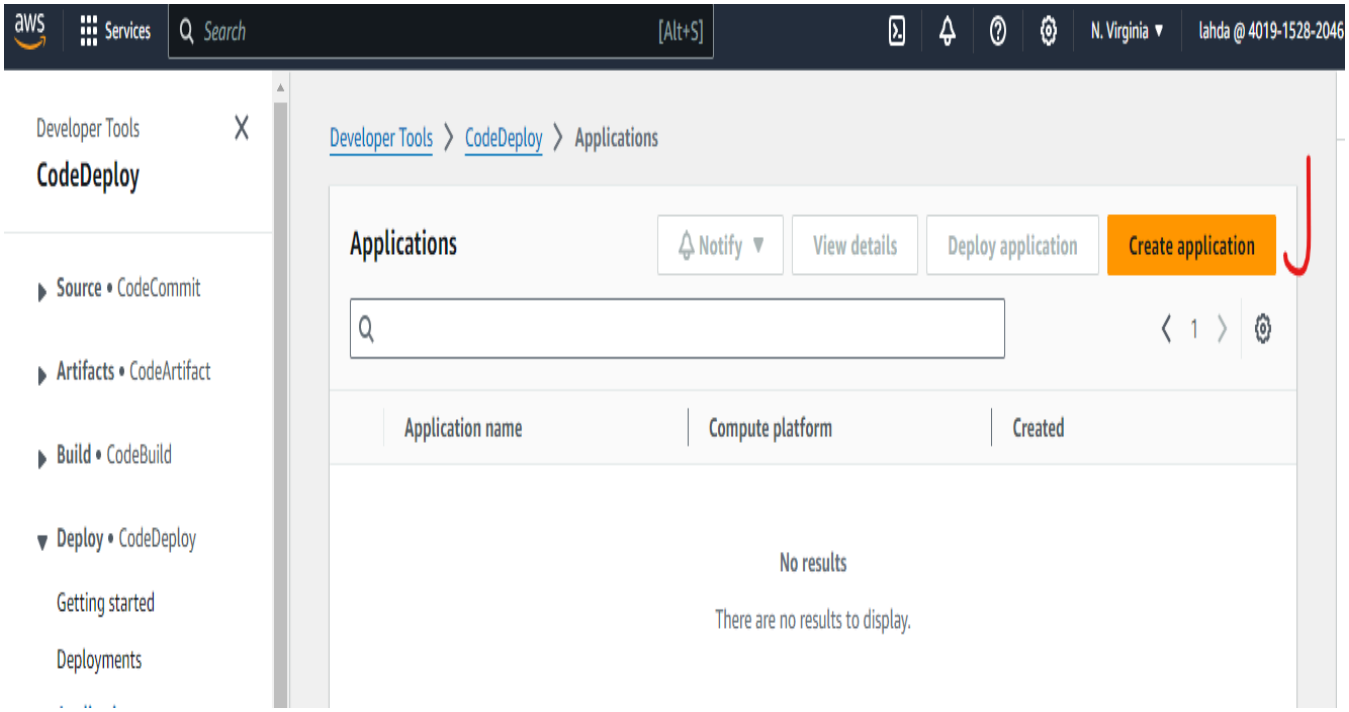
10. Choisissez Launch instance (Lancer une instance).

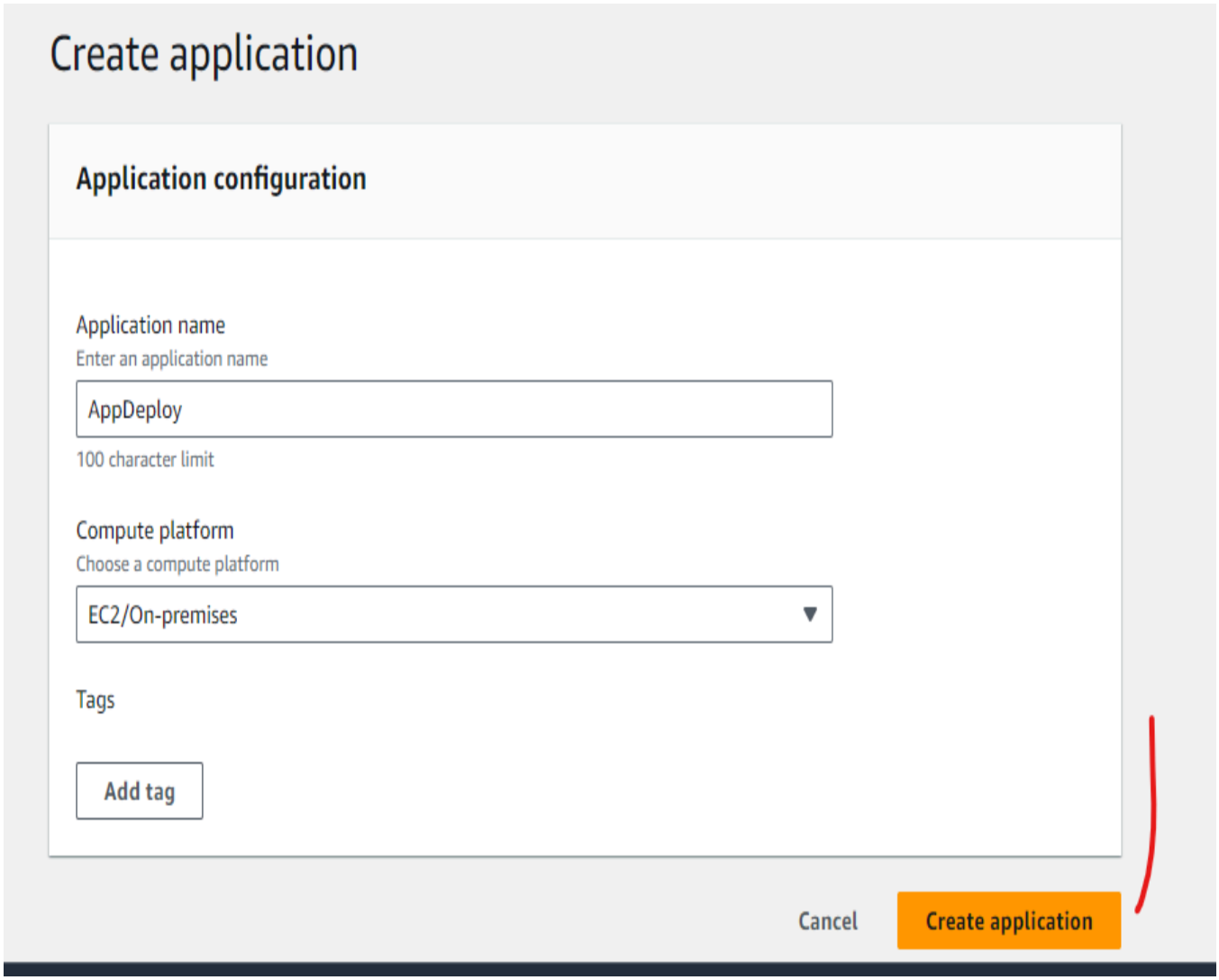
Etape 5 : création d'un déploiement dans CodeDeploy



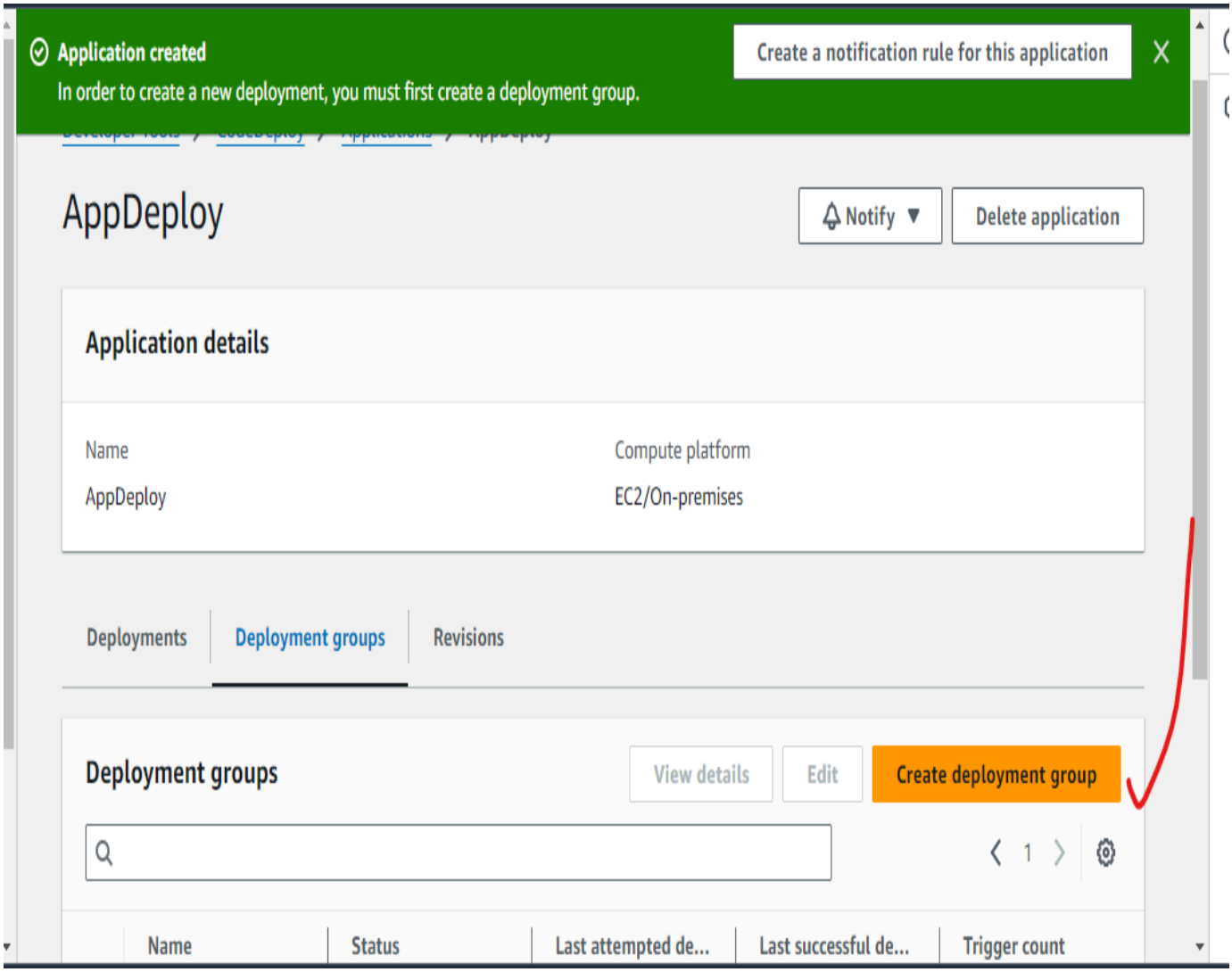
- Création d'une application



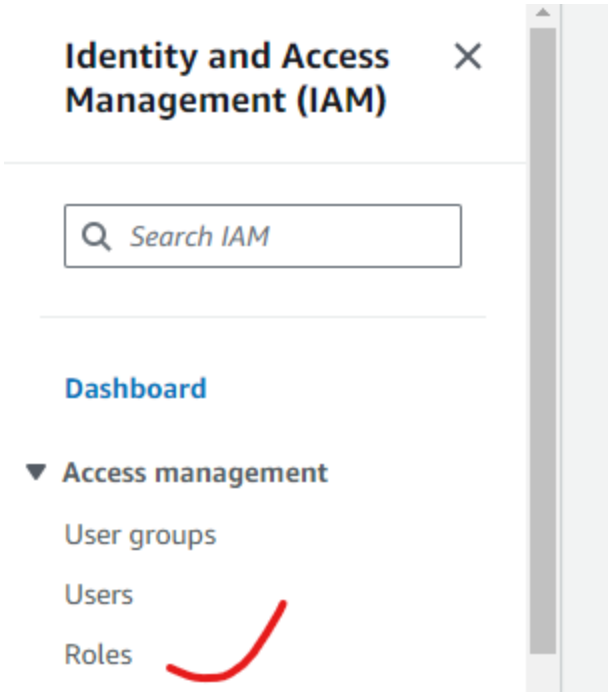

aws
Services
Search
[Alt+S]
N. Virginia
lahda @ 4019-1528-2046
Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Developer Tools > CodeDeploy > Applications
Applications
Notify
View details
Deploy application
Create application
1
Application name
Compute platform
Created
No results
There are no results to display.

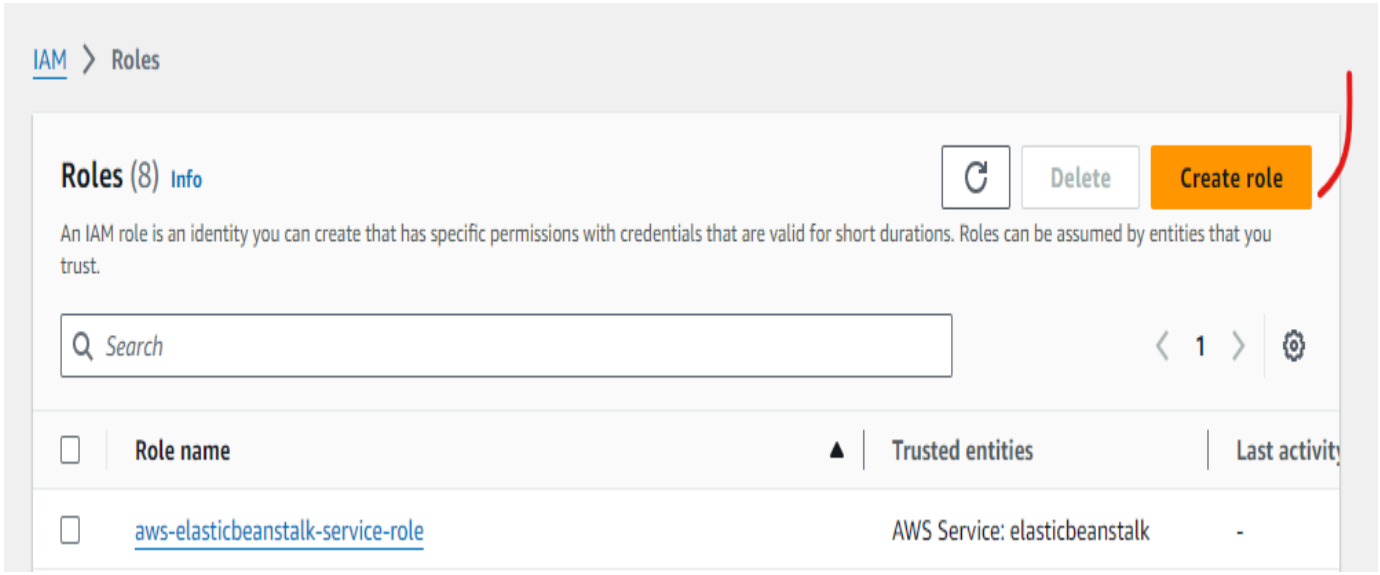


- Création de déploiement

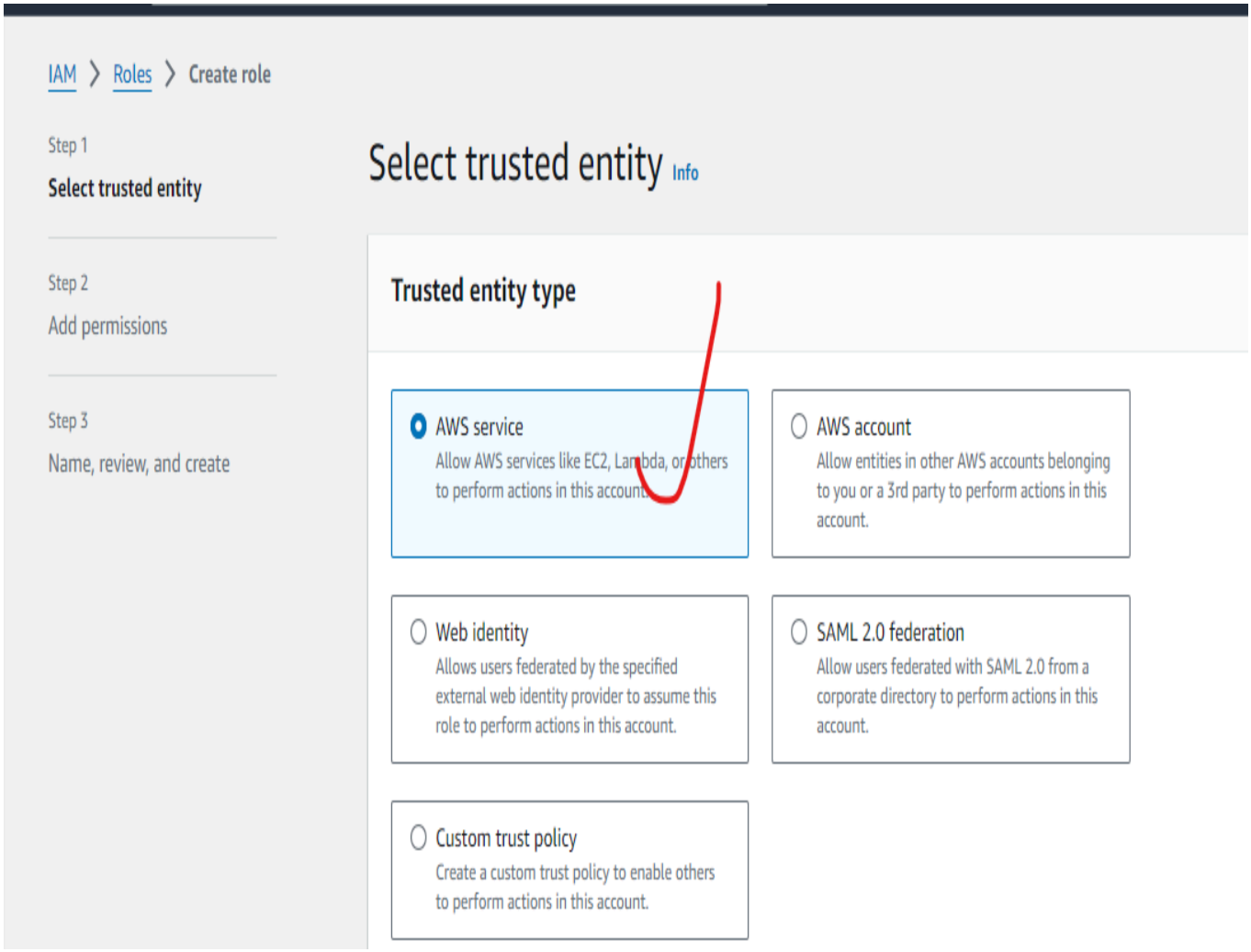


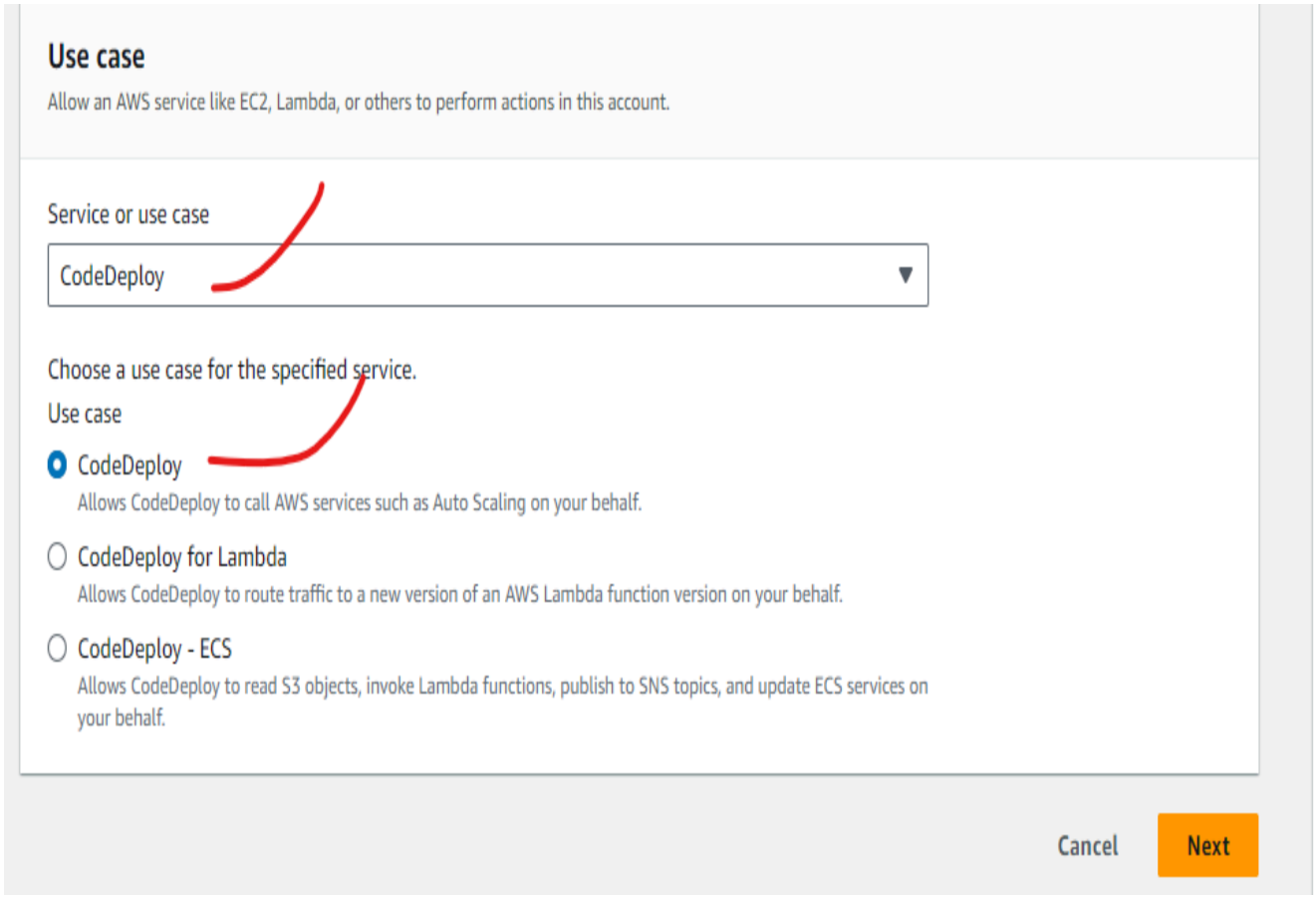
- **Creation de rôle IAM pour Codedeploy**

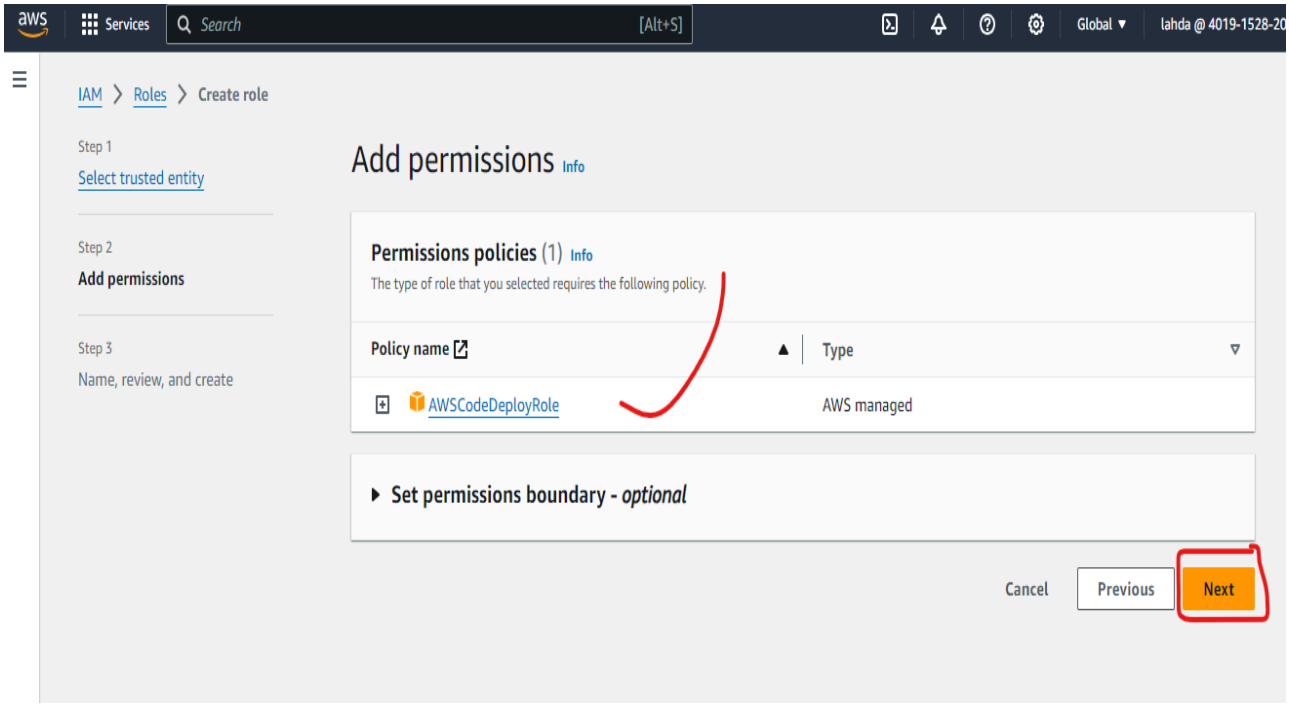




Project proposed by Lahda Biassou Alphonsine

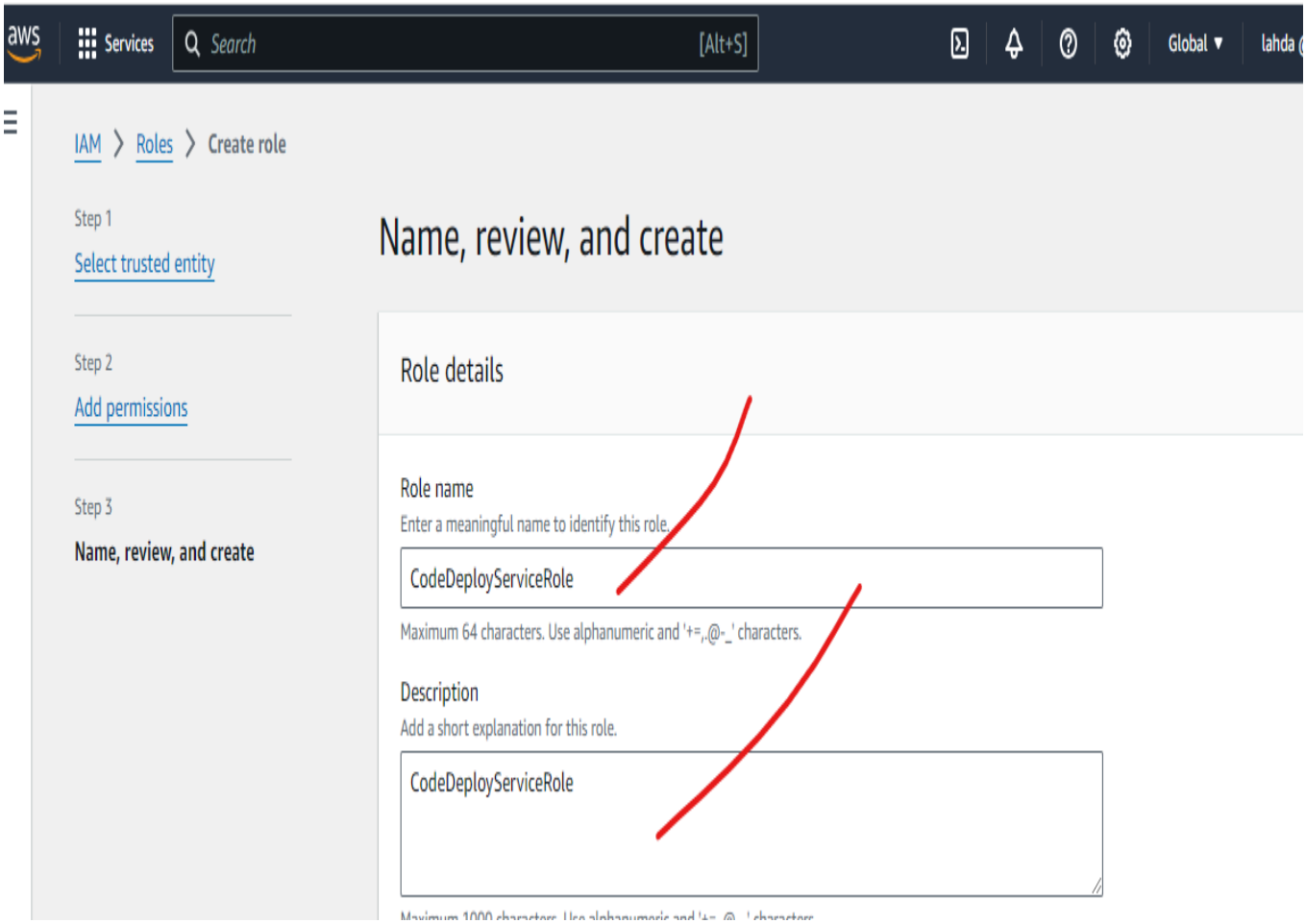

IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts 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.

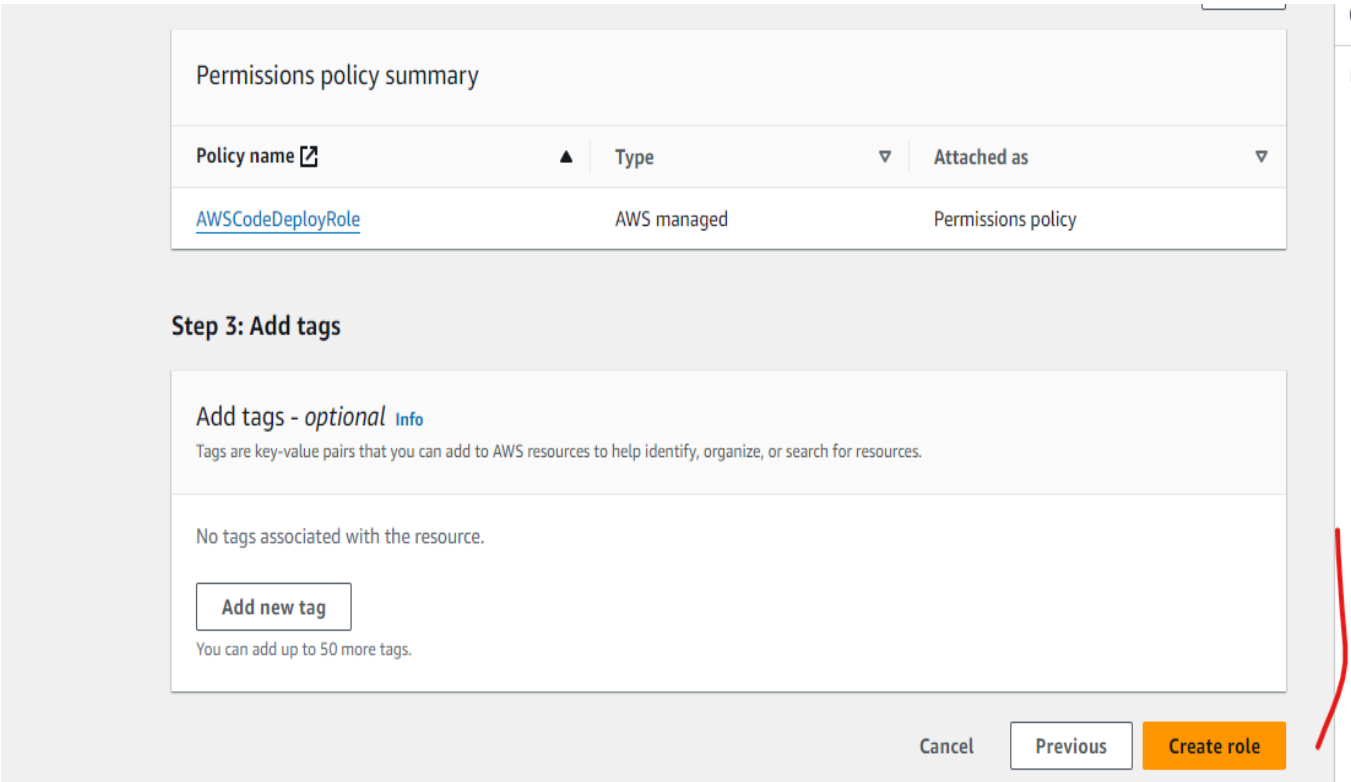




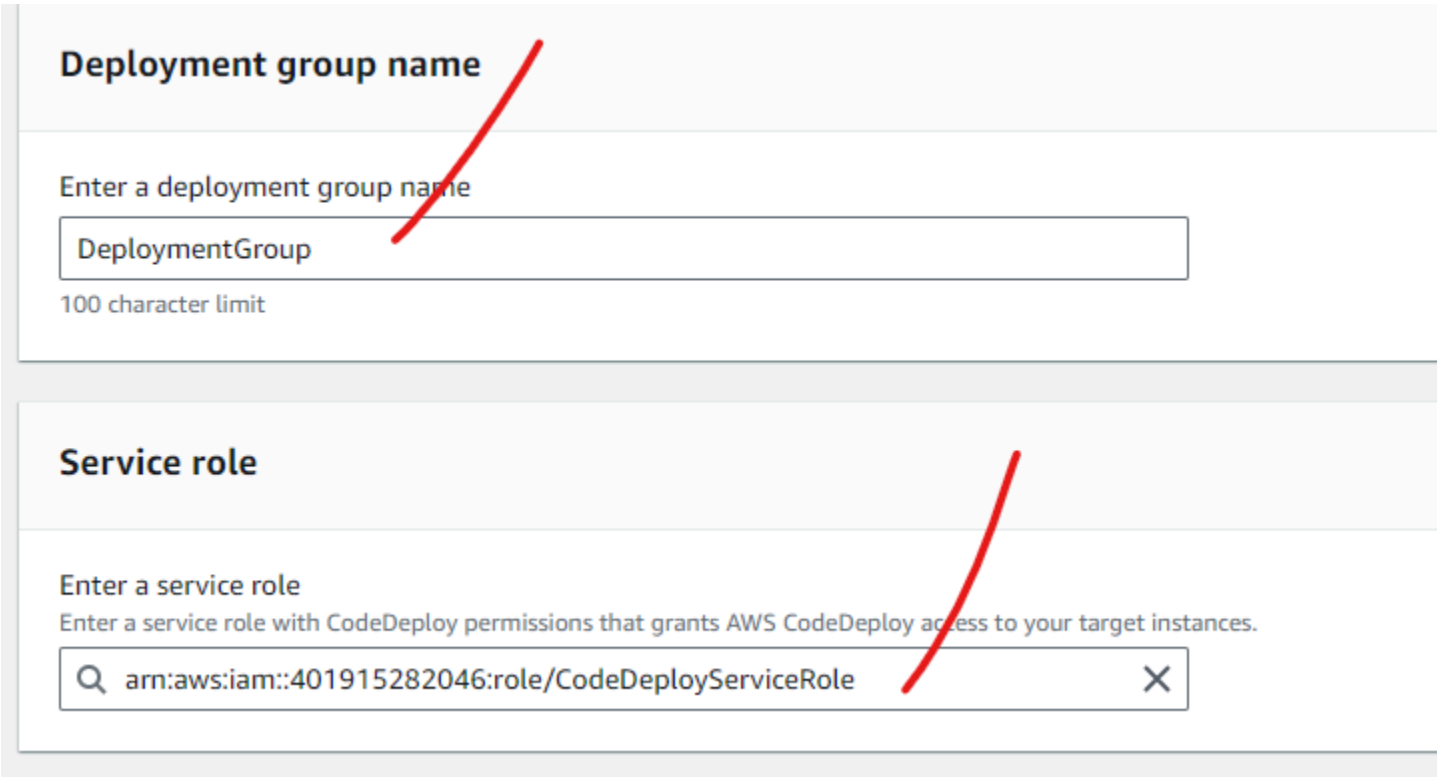
Project proposed by Lahda Biassou Alphonsine

- Spécification de nom du rôle et création

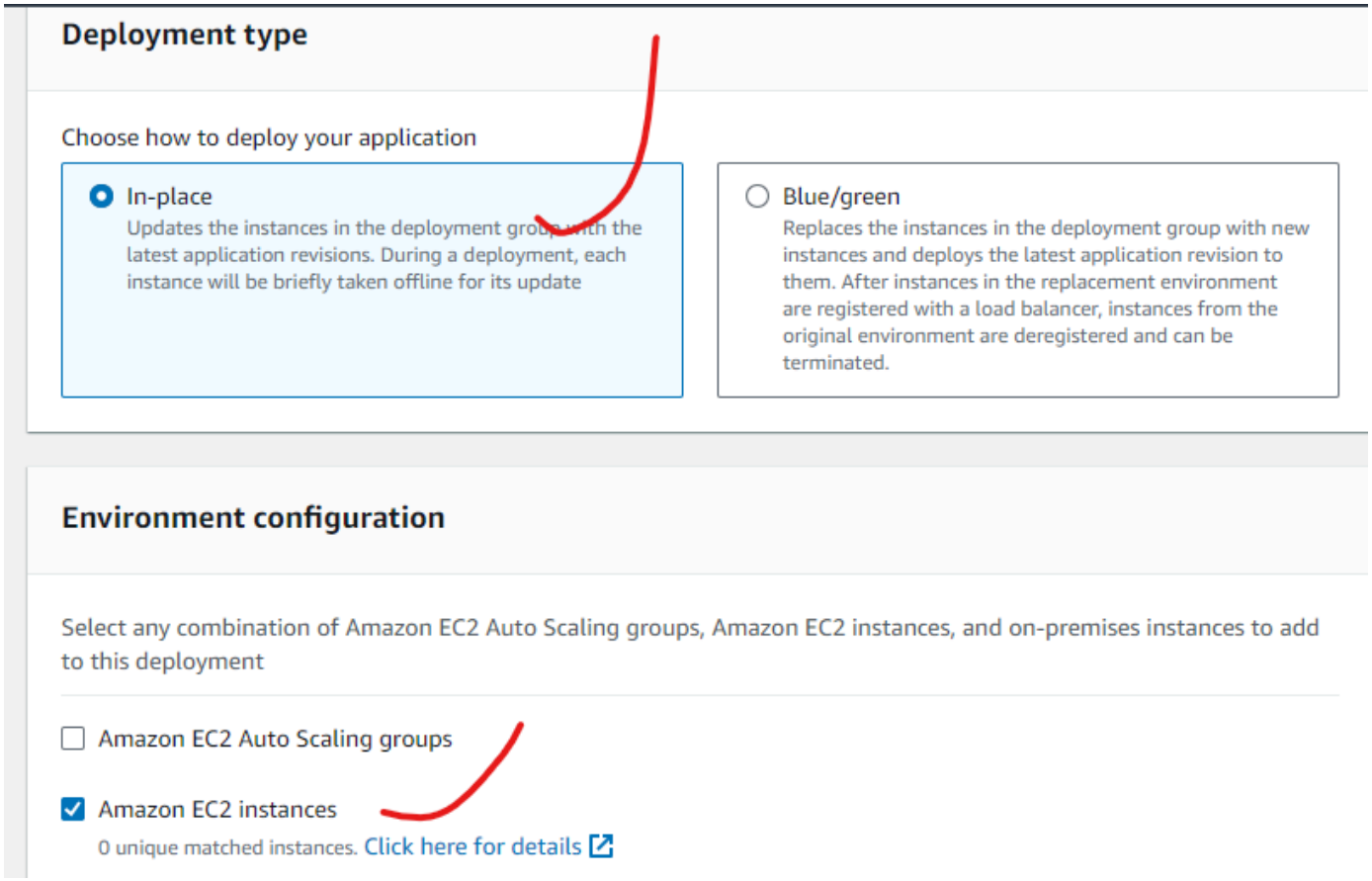


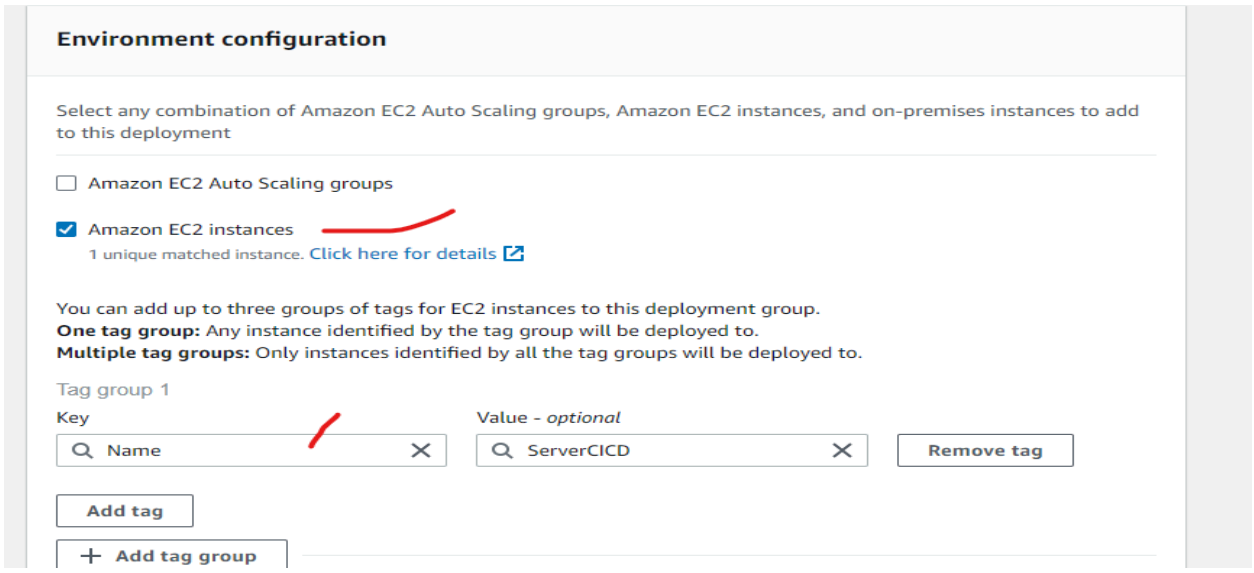


- **Création de groupe de déploiement**

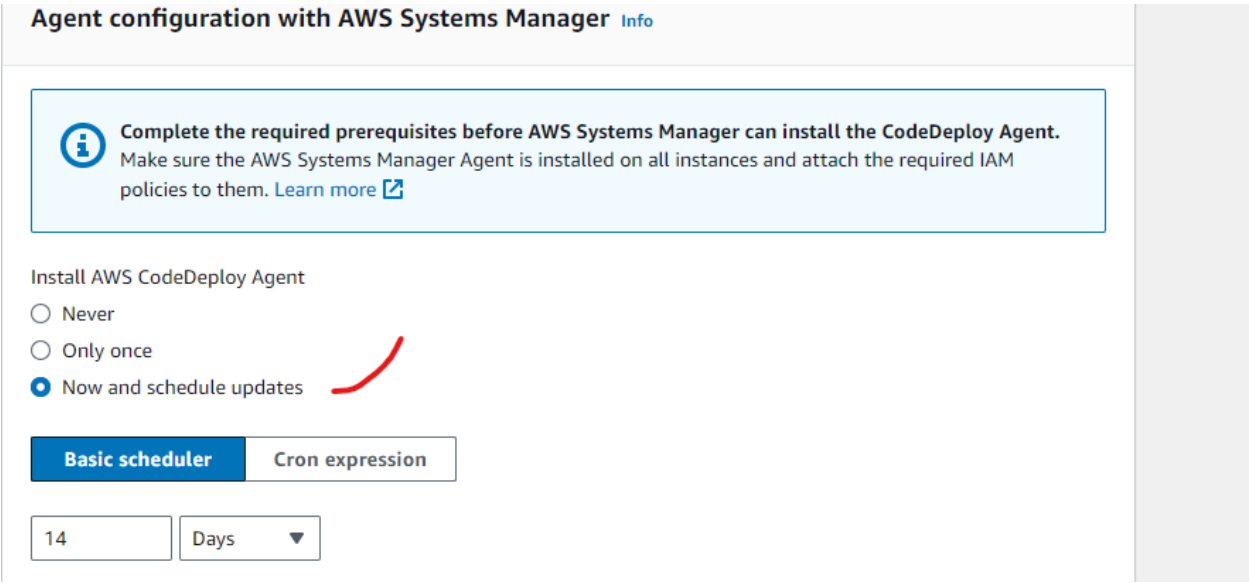


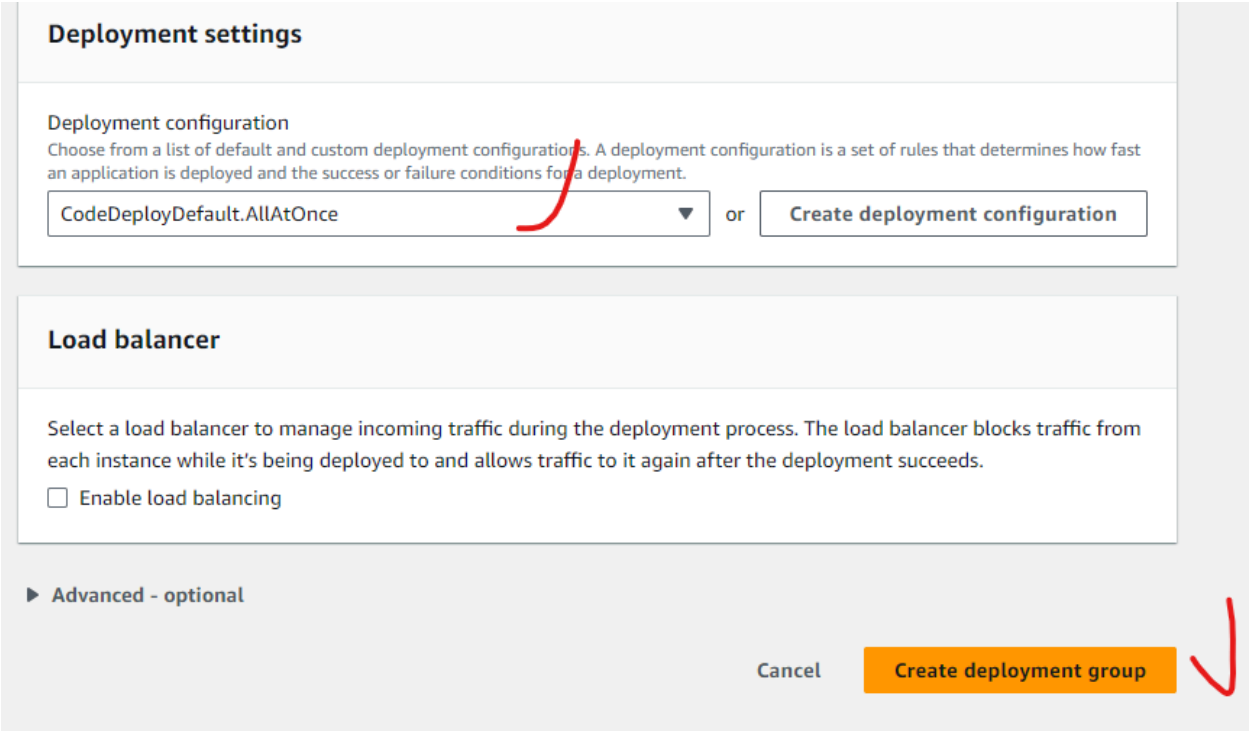
Project proposed by Lahda Biassou Alphonsine



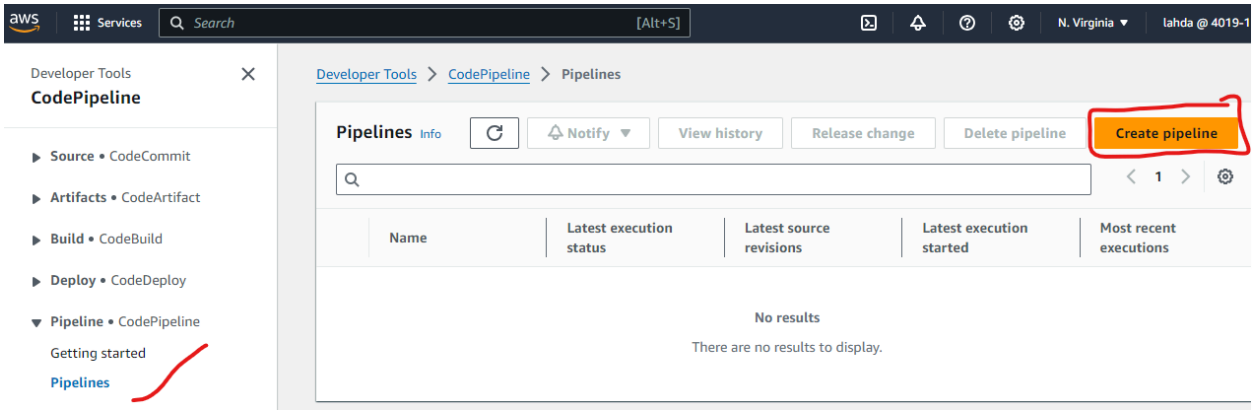


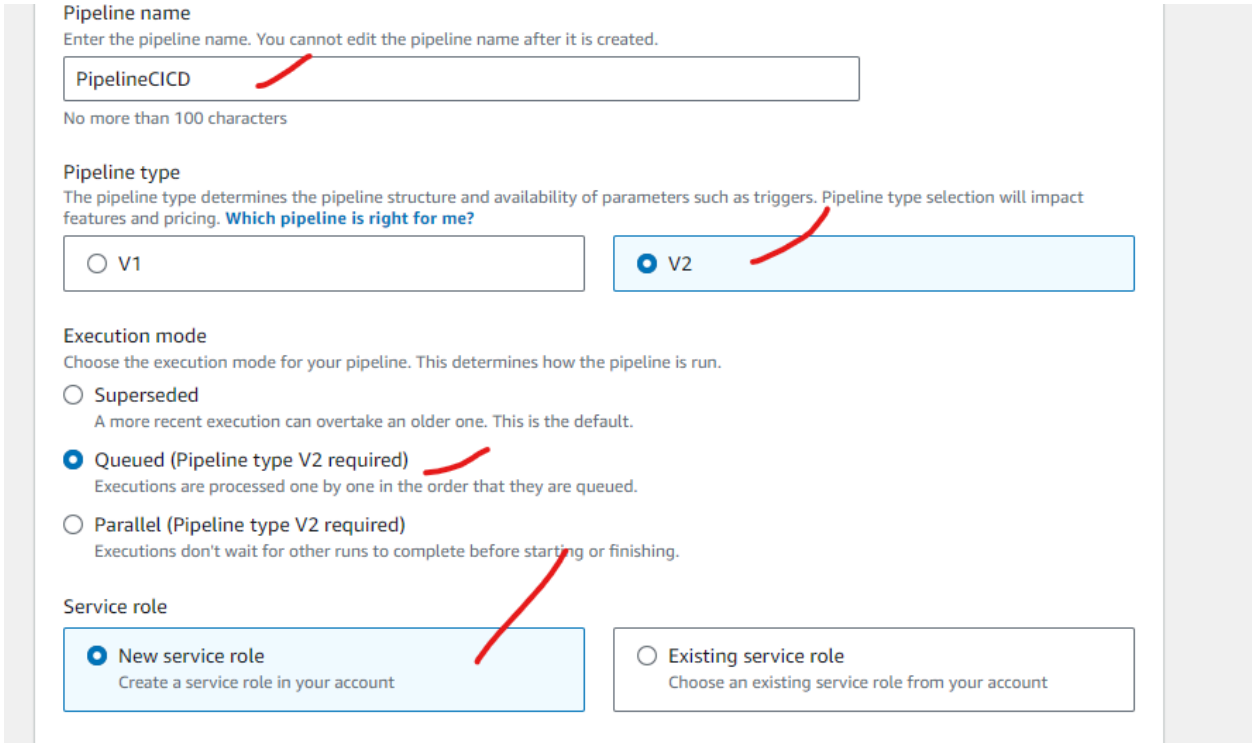
Project proposed by Lahda Biassou Alphonsine



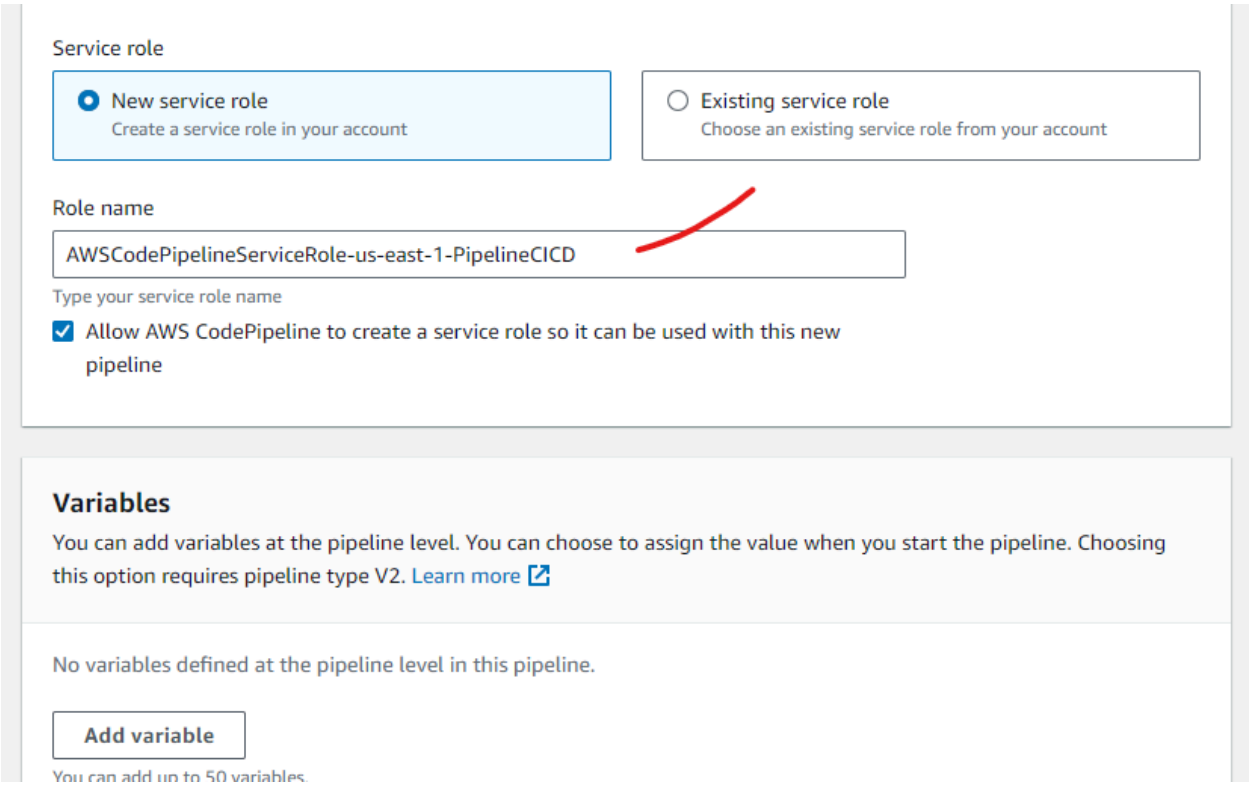


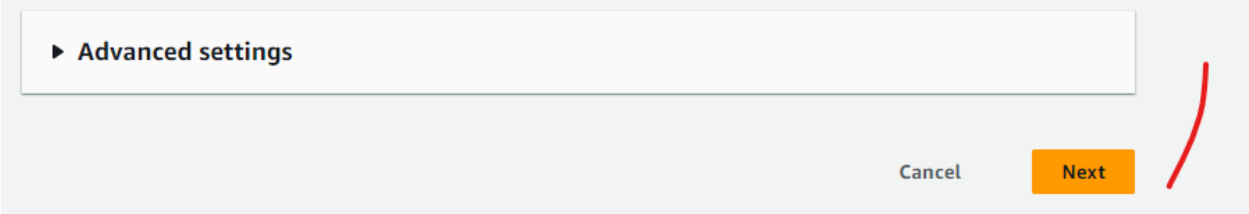
Étape 6: création et configuration du pipeline avec CodePipeline

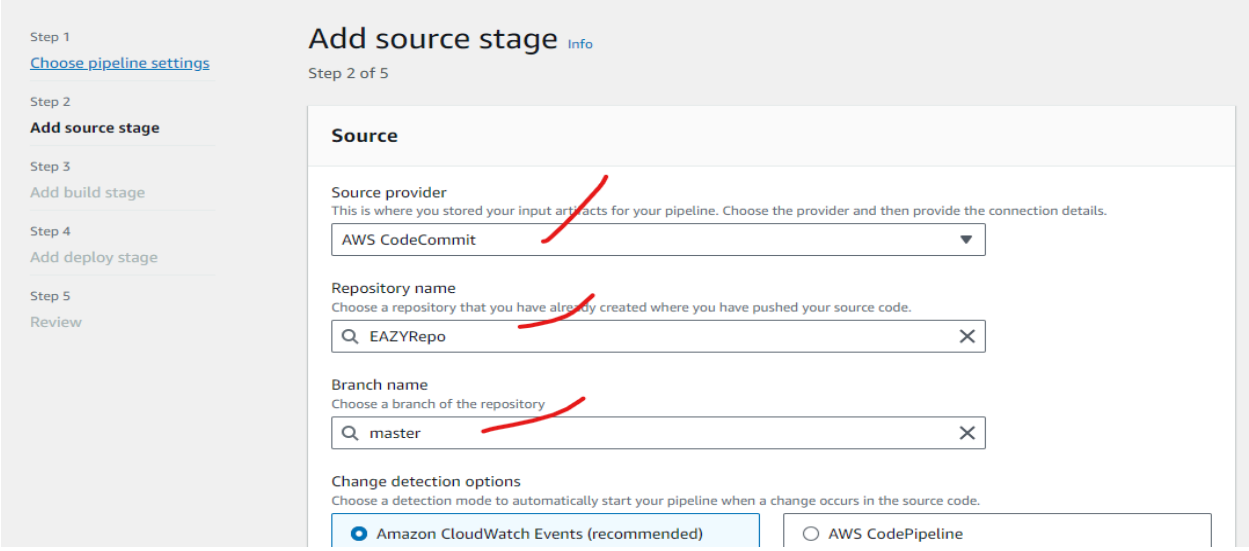




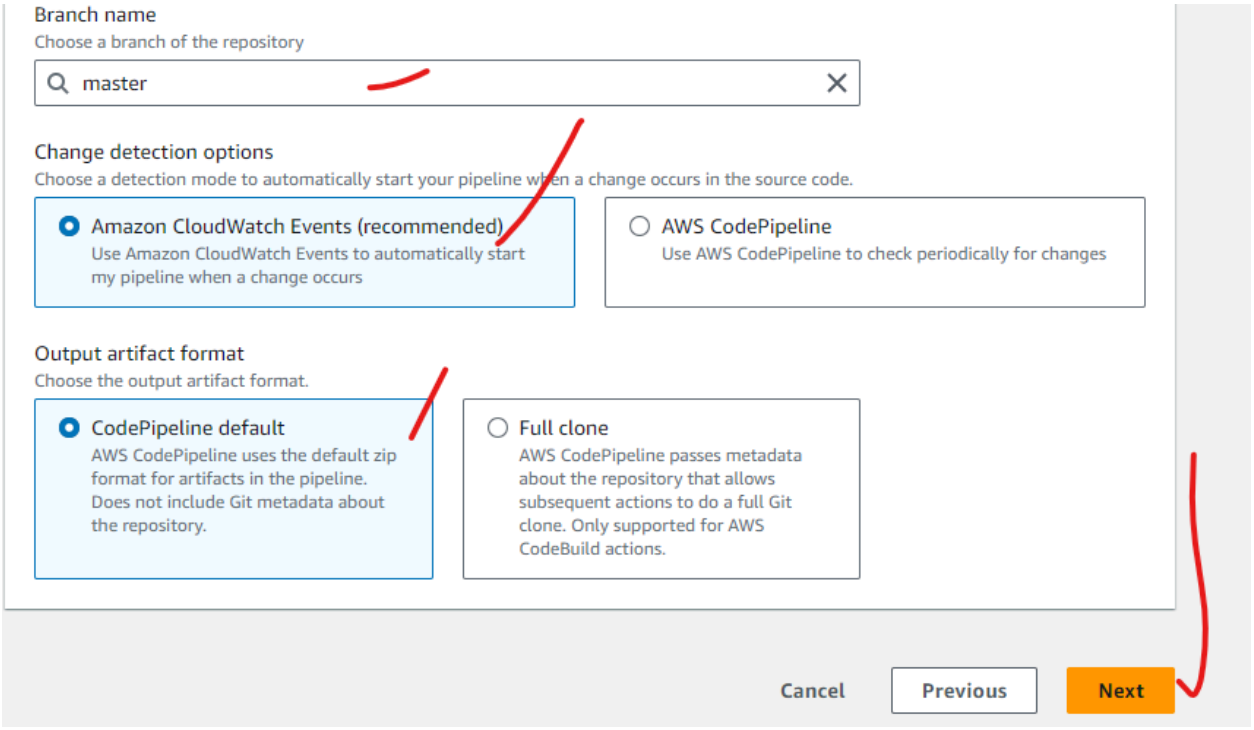
Project proposed by Lahda Biassou Alphonsine

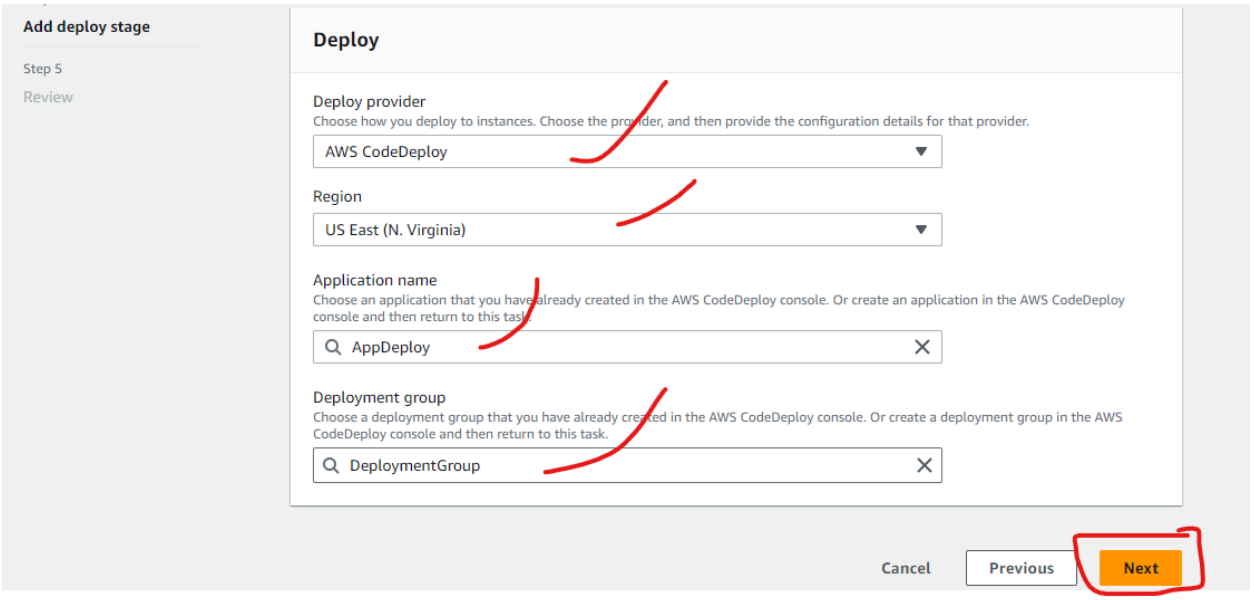




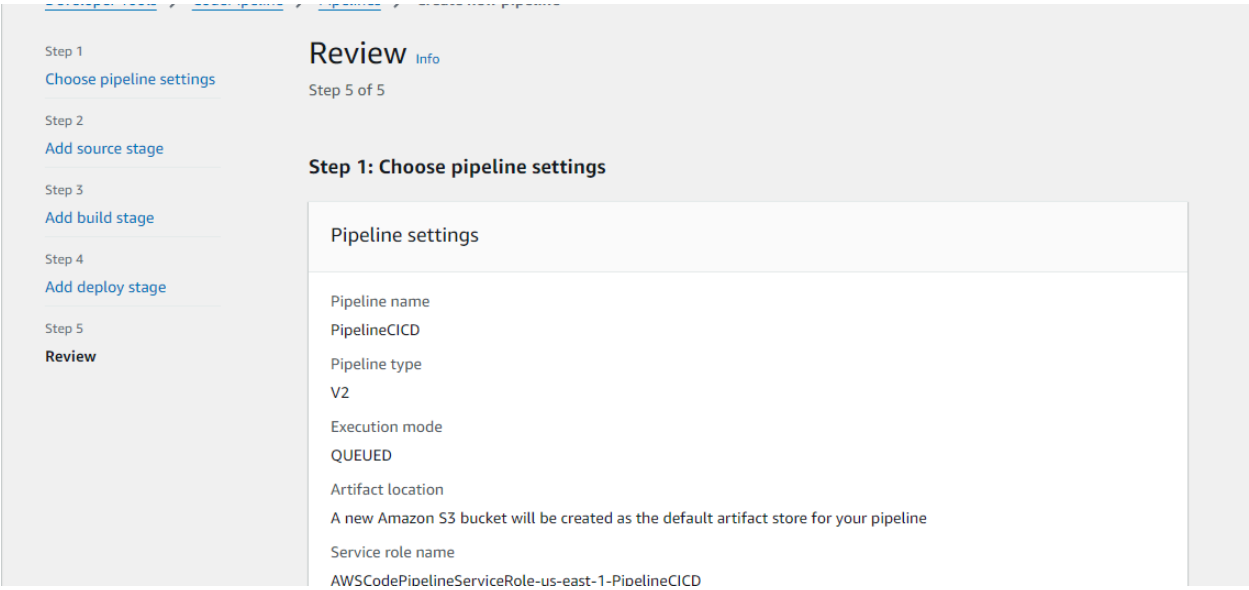


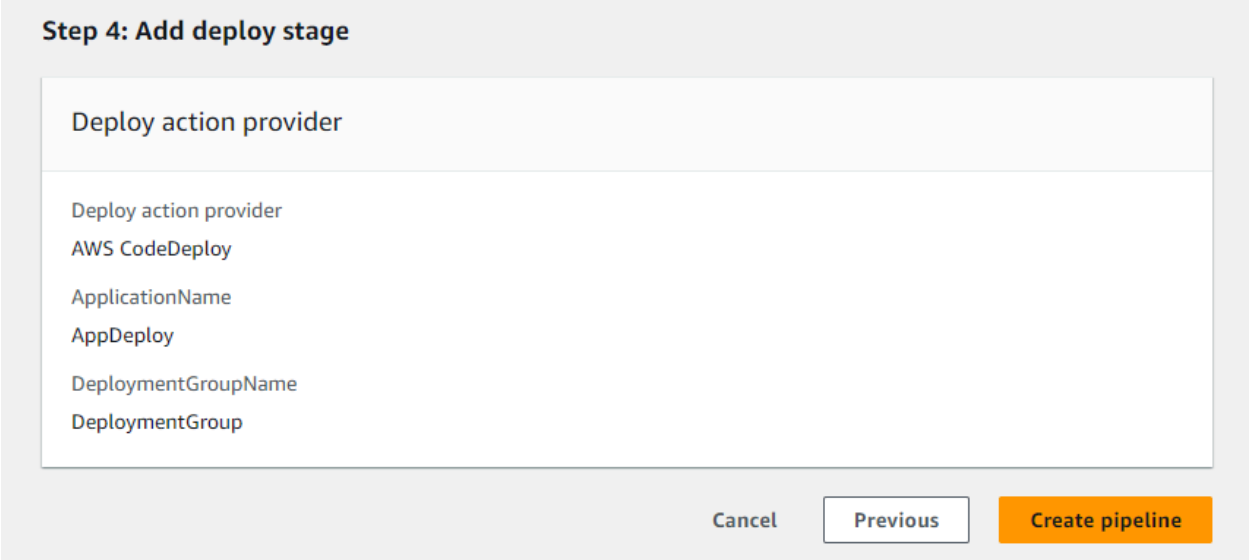
Project proposed by Lahda Biassou Alphonsine

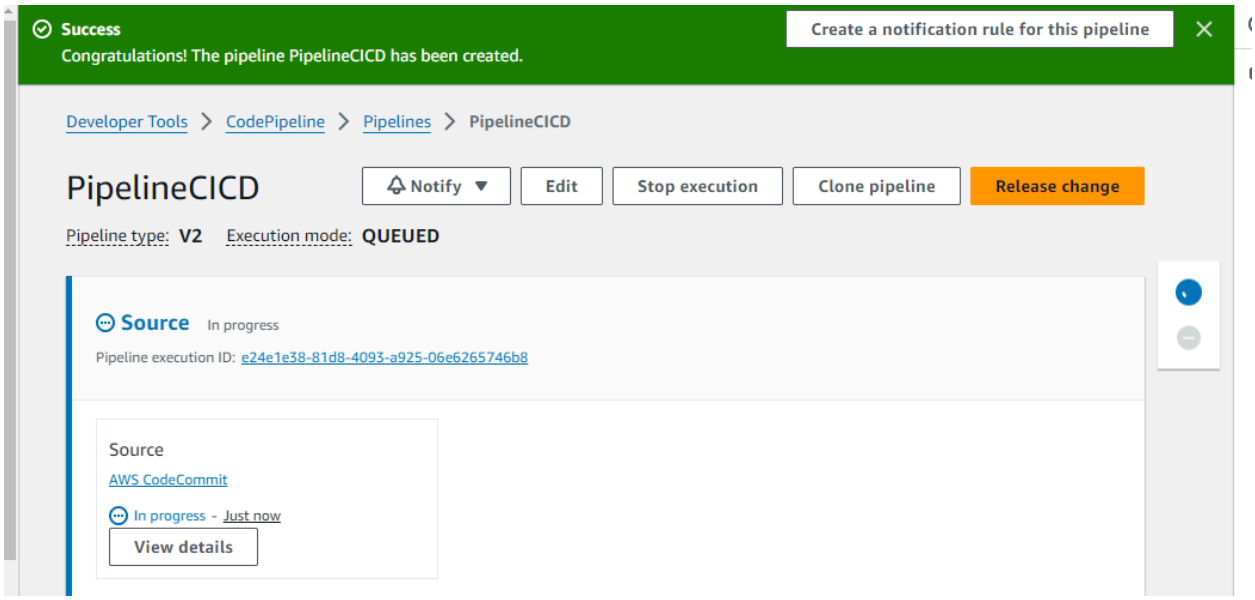


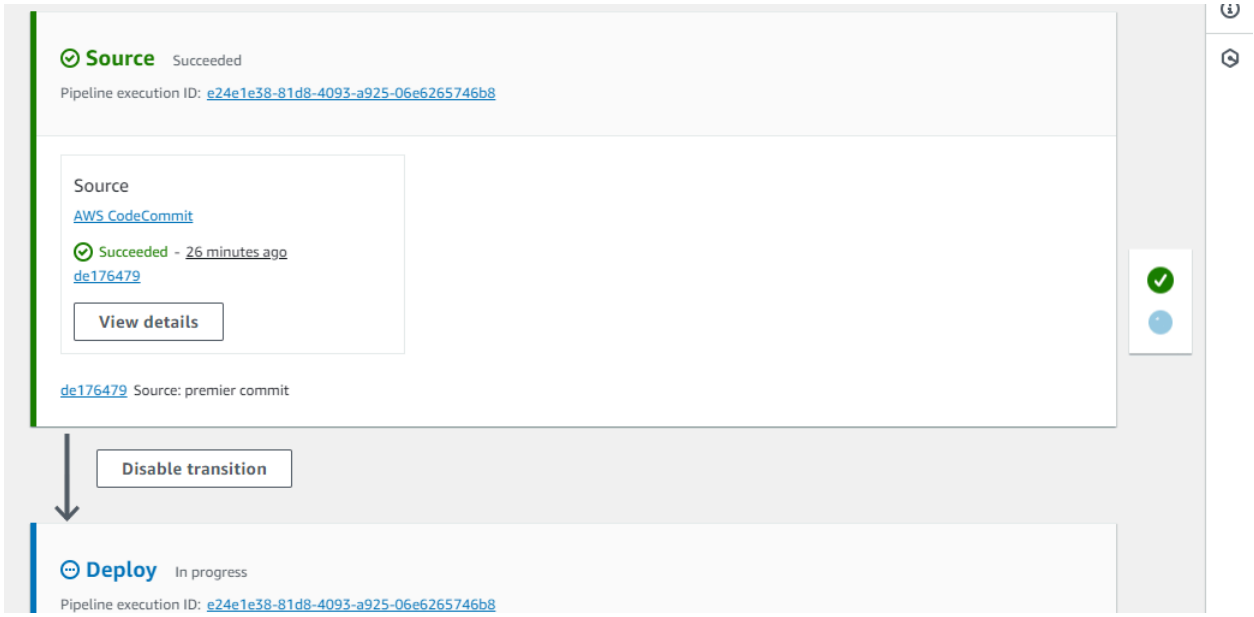


Project proposed by Lahda Biassou Alphonsine


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.