# Configurer un pipeline de déploiement en continu avec beanstalk en blue green (dans ce cas on utilisera github, le principe est le même si on utilise codecommit)

# Étape 1 : Création d'un environnement de déploiement

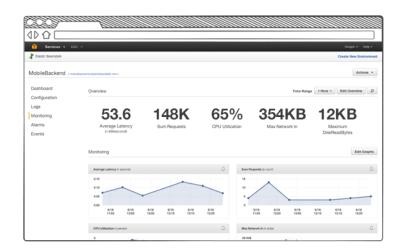
Votre pipeline de déploiement continu aura besoin d'un environnement cible contenant des serveurs virtuels ou des instances Amazon EC2, dans lequel l'exemple de code sera déployé. Vous préparerez cet environnement avant de créer le pipeline.

a. Pour simplifier la procédure d'installation et de configuration d'instances EC2 pour ce didacticiel, vous lancerez un exemple d'environnement grâce à AWS Elastic Beanstalk. Elastic Beanstalk permet d'héberger vos applications web en toute simplicité sans avoir à lancer, configurer ou faire fonctionner des serveurs virtuels vous-même. Il met en service et pilote automatiquement l'infrastructure (par ex. des serveurs virtuels, des équilibreurs de charge, etc.) et vous fournit la pile d'application (par ex. système d'exploitation, langue et infrastructure, serveur web et d'applications, etc.).

b. Choisissez *PHP* dans le menu déroulant, puis cliquez sur Launch Now.

Remarque: Si vous avez déjà créé une application Elastic Beanstalk, cliquez sur: Create New Application dans l'angle supérieur droit. Attribuez un nom à l'application et créez un *environnement de serveur web*. Sélectionnez *PHP* comme plateforme et *Single Instance* comme type d'environnement. Si vous prévoyez de vous connecter à distance à vos instances, sélectionnez une paire de clés. Dans le cas contraire, conservez les valeurs par défaut pour les autres options et créez l'environnement pour votre pipeline de déploiement continu.





#### Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can deploy, monitor, and scale an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

To deploy your existing web application, create an application source bundle and then create a new application. If you're using Git and would prefer to use it with our command line tool, please see Getting Started with the EB CLI.

To deploy a sample application with just one click, select a platform and click Launch Now.

By launching the sample application, you allow AWS Elastic Beanstalk to administer AWS resources and necessary permissions on your behalf. Learn more.

PHP Looking for a different platform? Let us know.

AWS Elastic Beanstalk will create an environment running PHP 5.6 on 64bit Amazon Linux 2016.03 v2.1.1. Change platform version.

Launch Now

#### Get Started in Three Easy Steps



Select a Platform



Upload an Application or Use a Sample



Run it!

Start Now by Selecting Your Platform









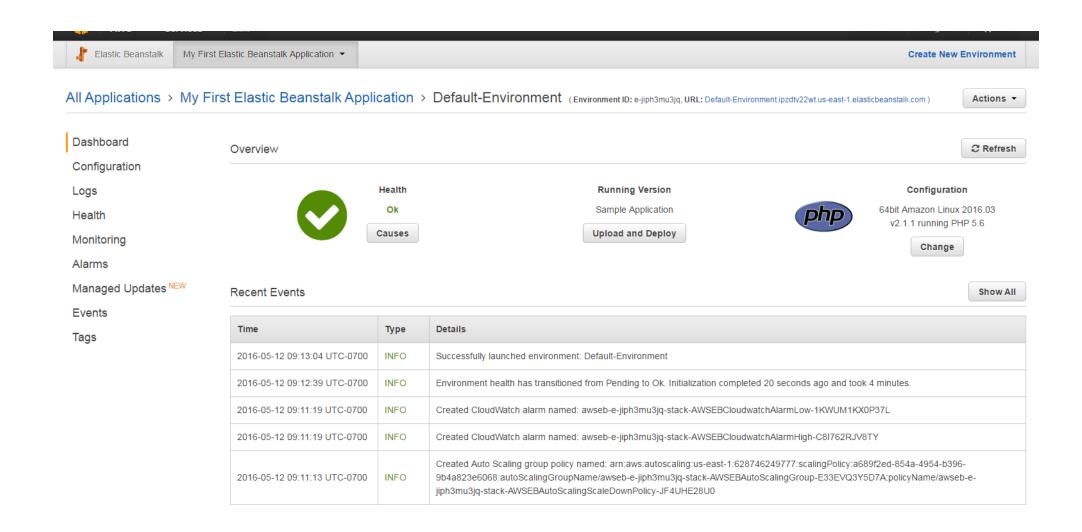




...and more

c. Elastic Beanstalk commencera à créer un exemple d'environnement dans lequel vous déploierez votre application. Il créera une instance Amazon EC2, un groupe de sécurité, un groupe Auto Scaling, un compartiment Amazon S3, des alarmes Amazon CloudWatch et un nom de domaine pour votre application.

Remarque : cette opération prendra plusieurs minutes



Cliquer sur action et dupliquez l'environnement, donner lui un autre nom

# Étape 2 : Obtention d'une copie de l'exemple de code

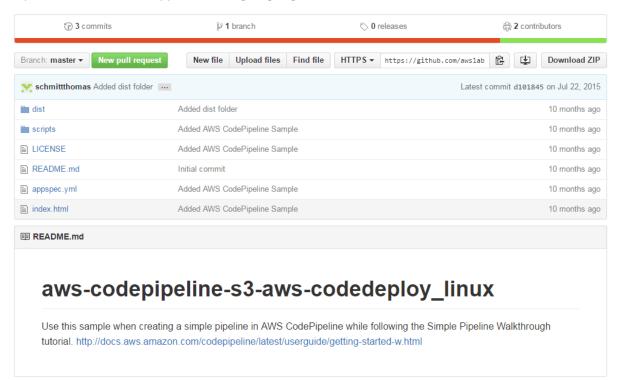
Au cours de cette étape, vous extrairez une copie du code de l'exemple d'application, puis vous choisirez une source pour héberger le code. Le pipeline prend le code de la source, puis effectue des actions dessus.

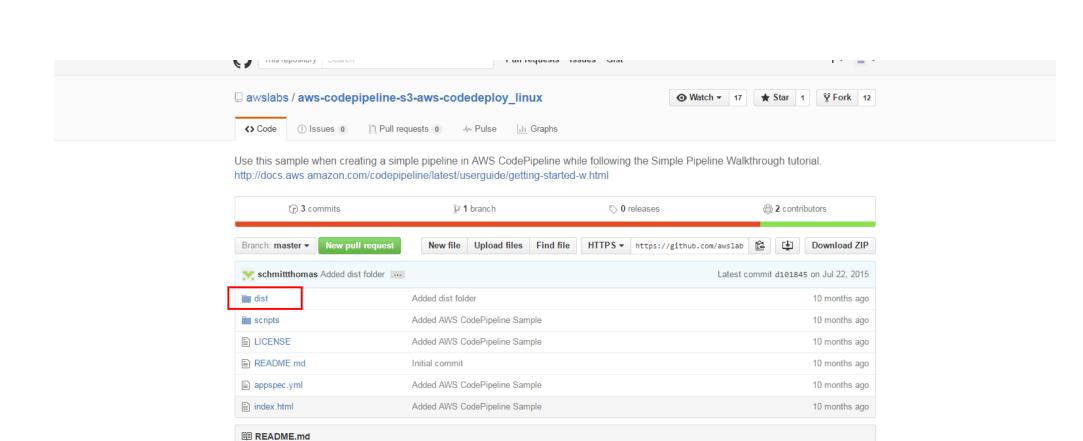
Vous pouvez utiliser l'une de ces trois options comme source : un référentiel GitHub, un compartiment Amazon S3 ou un référentiel AWS CodeCommit. Sélectionnez votre préférence et appliquez la procédure ci-après :

- a. Si vous souhaitez utiliser votre compte GitHub:
  - Accédez à notre référentiel GitHub contenant l'exemple de code sur le site <a href="https://github.com/awslabs/aws-codepipeline-s3-aws-codedeploy\_linux">https://github.com/awslabs/aws-codepipeline-s3-aws-codedeploy\_linux</a>.
  - Dupliquez la copie du référentiel dans votre propre compte GitHub en cliquant sur le bouton Fork dans l'angle supérieur droit.



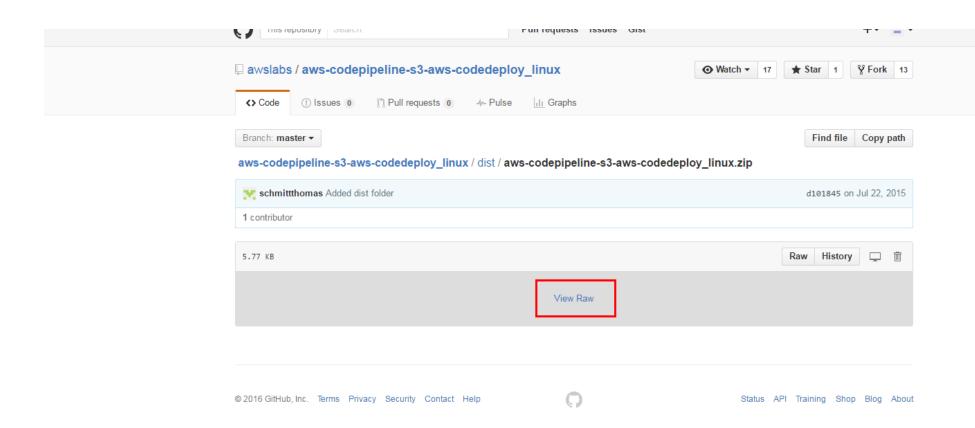
Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial. http://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-w.html

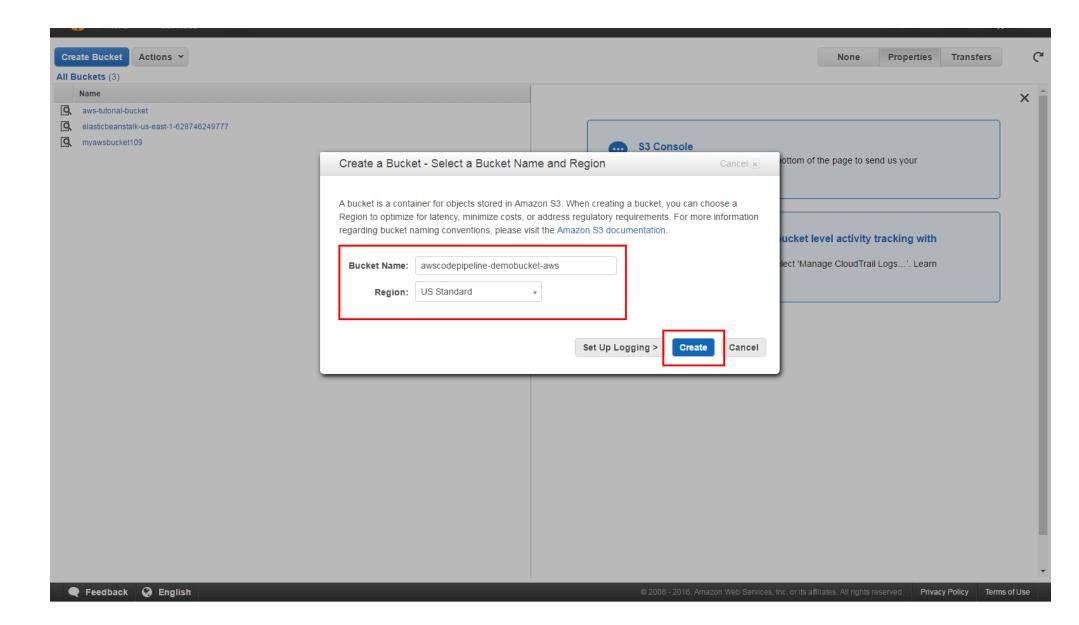


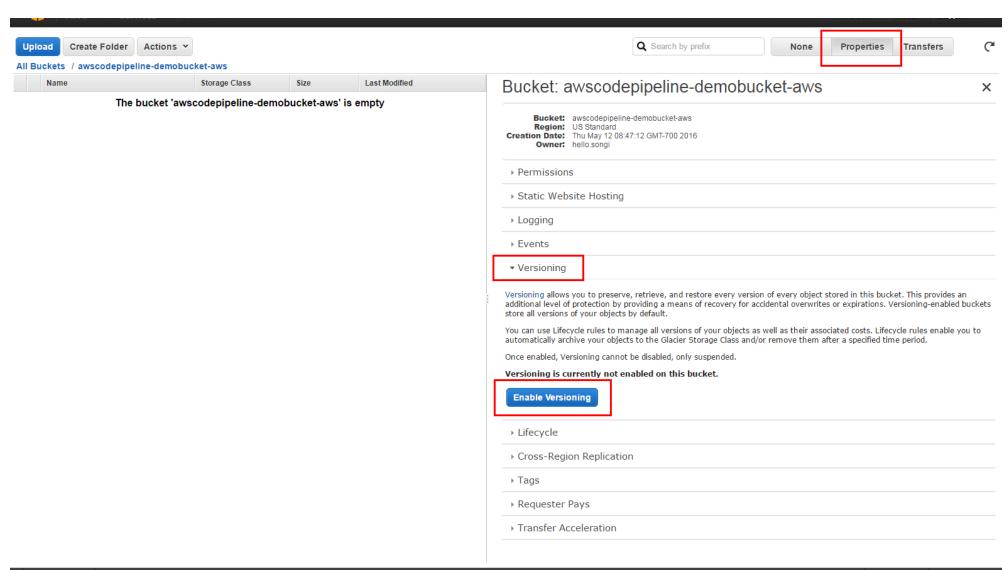


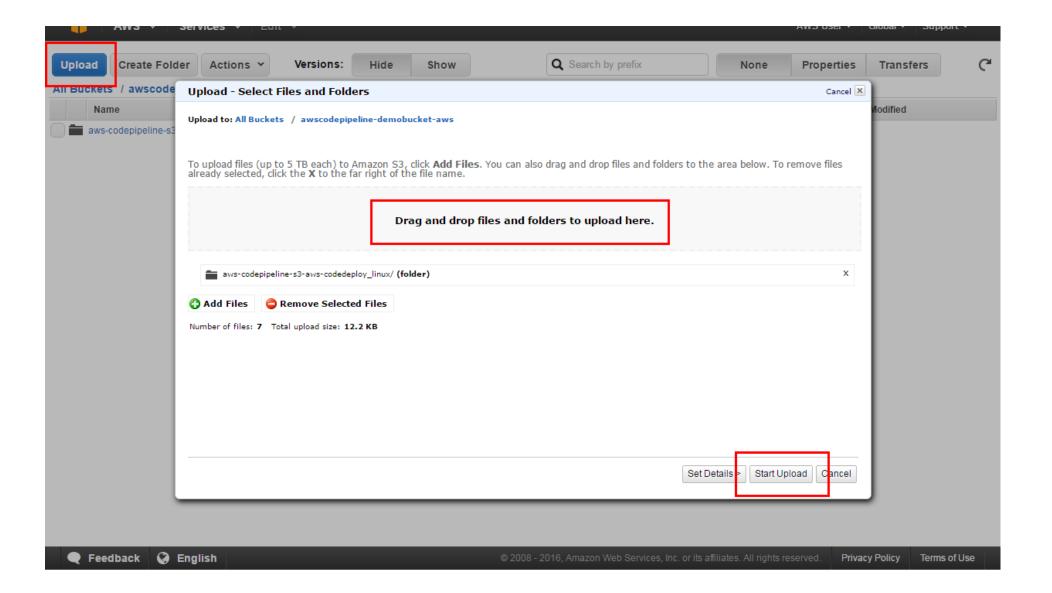
# aws-codepipeline-s3-aws-codedeploy\_linux

Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial. http://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-w.html



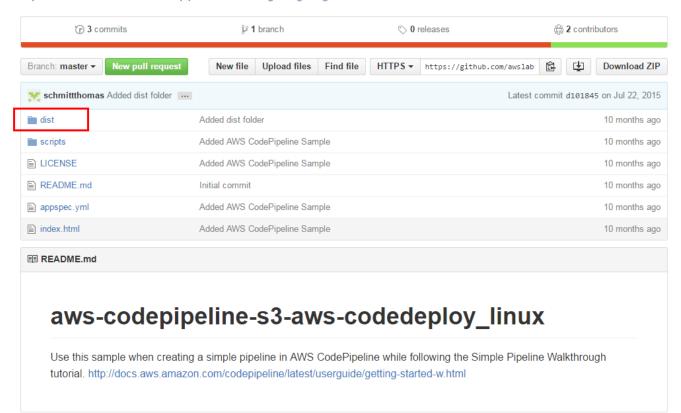


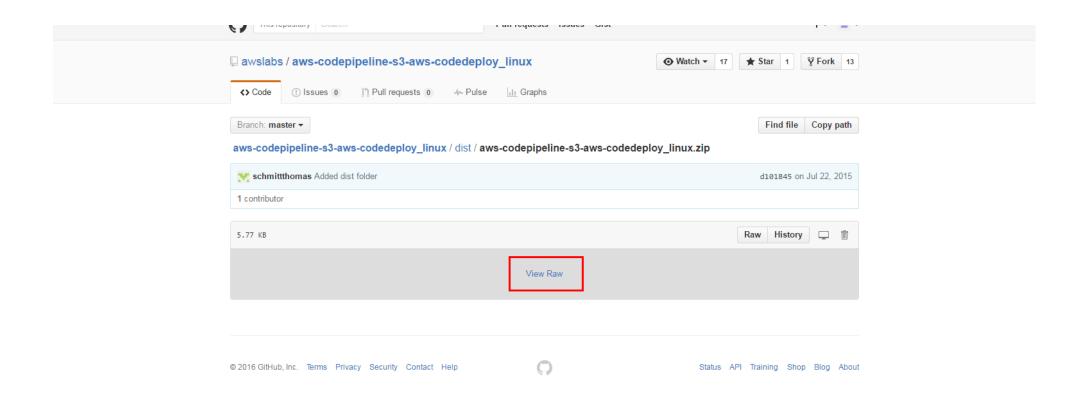






Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial. http://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-w.html







Create secure repositories for sharing your code in the cloud.

**Get started** 



#### Create

Quickly create a secure repository in the cloud with just a few steps.

Learn more



#### Connect

Push and pull commits between your local Git™ repo and your AWS CodeCommit repository.

Learn more



#### **Share**

Provide secure access to your repository to other users.

Learn more

AWS CodeCommit documentation & support

Getting started guide | Documentation | Support | Forums





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.

#### Access to the repository

Users connecting to an AWS CodeCommit repository for the first time must complete setup steps before they can use it. Learn more

Repository name\* PipelineRepo Description 1000 character limit

\*Required

Cancel

Create repository

# Étape 3 : Créer votre pipeline

Au cours de cette étape, vous allez créer et configurer un pipeline simple avec deux actions : source et déployer. Vous fournirez CodePipeline avec les emplacements de votre référentiel source et de votre environnement de déploiement.

### a. Cliquez ici pour ouvrir la console AWS CodePipeline.

- Sur la page Welcome, cliquez sur Create pipeline.
- Si vous utilisez AWS CodePipeline pour la première fois, une page d'introduction s'affiche au lieu de la page Welcome. Cliquez sur Get started.



# AWS CodePipeline

Visualize and automate the different stages of your software release process, and watch your code go.

**Get started** 



### Automate

Build, test, and deploy your code, your way.

## Track

Know how your code is doing at every step of the release.



#### Learn

Learn more about how pipelines work and how you can use them.

Learn more Learn more Learn more

AWS CodePipeline Documentation & Support

Getting Started | Documentation | AWS Support | AWS CodePipeline Forums

Feedback Senglish

Privacy Policy

Terms of Use

(cliquez pour zoomer)

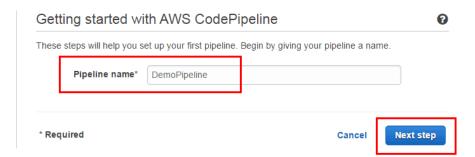
# b. Sur la page Step 1: Name :

- Pipeline name : entrez le nom pour votre pipeline, DemoPipeline.
- Cliquez sur Next Step.

Remarque : Une fois le pipeline créé, vous ne pouvez plus modifier son nom.

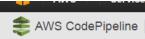


Step 1: Name Step 2: Source Step 3: Build Step 4: Beta Step 5: Service Role Step 6: Review



c. Sur la page *Step 2: Source*, sélectionnez l'emplacement de la source que vous avez sélectionnée et suivez la procédure ci-après :

- Fournisseur de source : GitHub
  - Dans la section Connect to GitHub, cliquez sur Connect to GitHub.
  - Une nouvelle fenêtre de navigateur s'ouvre, qui vous permet de vous connecter à GitHub. Si vous y êtes invité, saisissez vos informations d'identification GitHub.
  - Vous serez invité à autoriser l'accès aux applications à votre compte. Choisissez Authorize application.



#### Step 1: Name

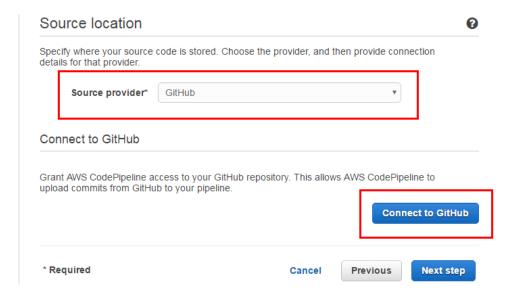
Step 2: Source

Step 3: Build

Step 4: Beta

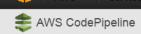
Step 5: Service Role

Step 6: Review



### Précisez le référentiel et la branche :

- Référentiel : dans la liste déroulante, choisissez le référentiel GitHub que vous souhaitez utiliser en tant qu'emplacement de la source pour votre pipeline. Cliquez sur le référentiel dupliqué dans votre compte GitHub contenant l'exemple de code nommé aws-codepipeline-s3-aws-codedeploy\_linux.
- Branch : choisissez dans la liste déroulante la branche que vous souhaitez utiliser, la principale.
- Cliquez sur Next Step.



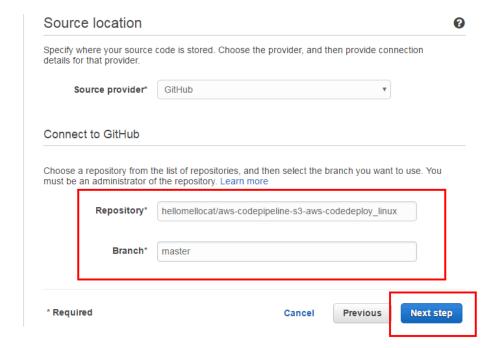
#### Step 1: Name Step 2: Source

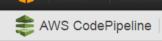
Step 3: Build

Step 4: Beta

Step 5: Service Role

Step 6: Review





#### Step 1: Name

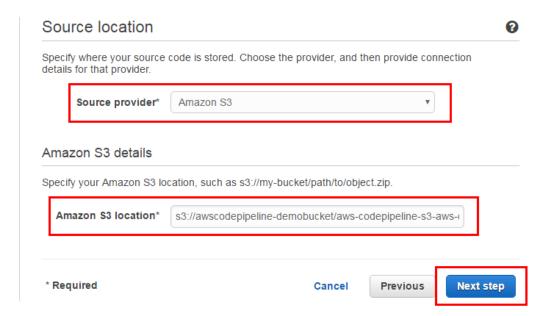
#### Step 2: Source

Step 3: Build

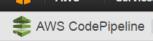
Step 4: Beta

Step 5: Service Role

Step 6: Review



Terms of l



#### Step 1: Name

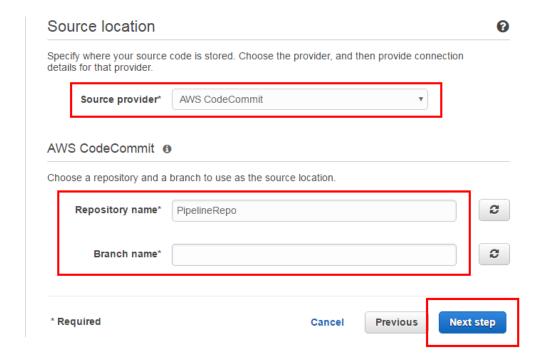
Step 2: Source

Step 3: Build

Step 4: Beta

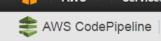
Step 5: Service Role

Step 6: Review



d. Un véritable pipeline de déploiement continu nécessite une étape de génération, au cours de laquelle le code est compilé et l'unité testée. CodePipeline vous permet de connecter votre fournisseur de génération à votre pipeline. Cependant, vous passez l'étape de génération au cours de ce didacticiel.

- Sur la page Step 3: Build, choisissez No Build.
- Cliquez sur Next Step.



Step 1: Name

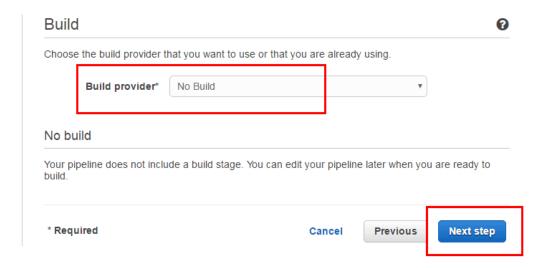
Step 2: Source

Step 3: Build

Step 4: Beta

Step 5: Service Role

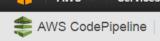
Step 6: Review



# e. Sur la page Step 4: Beta:

- Deployment provider : cliquez sur AWS Elastic Beanstalk.
- Application name: cliquez sur My First Elastic Beanstalk Application.
- Environment name : cliquez sur Default-Environment.
- Cliquez sur Next Step.

Remarque : Le nom « Bêta » est simplement le nom donné par défaut à cette étape du pipeline, tout comme « Source » était le nom donné à la première étape du pipeline.



Step 1: Name

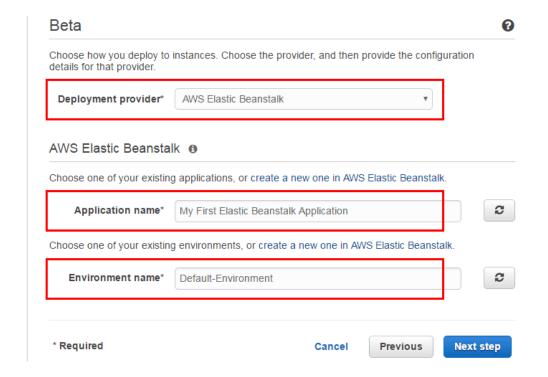
Step 2: Source

Step 3: Build

Step 4: Beta

Step 5: Service Role

Step 6: Review

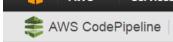


Privacy Policy

### f. Sur la page Step 5: Service Role:

- Service Role : cliquez sur Create role.
- Vous serez redirigé vers une page de la console IAM qui décrit le rôle AWS-CodePipeline-Service qui sera créé.
   Cliquez sur Allow
- Après avoir créé le rôle, vous êtes redirigé vers la page *Step 5: Service Role*, où *AWS-CodePipeline-Service* apparaît dans Role name. Cliquez sur Next Step.

Remarque: La création d'un rôle de service n'est requise que la première fois où vous créez un pipeline dans AWS CodePipeline. Si un rôle de service a déjà été créé, vous pourrez le choisir dans le menu déroulant des rôles. Etant donné que la liste déroulante affiche tous les rôles de service IAM associés à votre compte, si vous choisissez un nom différent de celui par défaut, assurez-vous qu'il soit reconnaissable en tant que rôle de service pour AWS CodePipeline.



Step 1: Name

Step 2: Source

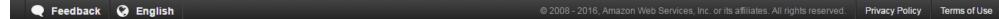
Step 3: Build

Step 4: Beta

Step 5: Service Role

Step 6: Review





Étape 4 : Activation de votre pipeline pour déployer votre code

Dans cette étape, vous allez lancer votre pipeline. Une fois votre pipeline créé, il s'exécute automatiquement. Dans un premier temps, il détecte l'exemple de code d'application dans votre emplacement source, il regroupe les fichiers, puis les transfère vers la deuxième étape que vous avez définie. Au cours de cette étape, il transmet le code à Elastic Beanstalk, qui contient l'instance EC2 qui hébergera votre code. Elastic Beanstalk gère le déploiement du code dans l'instance EC2.

a. Sur la page Step 6: Review, vérifiez les informations puis cliquez sur Create pipeline.



Step 1: Name

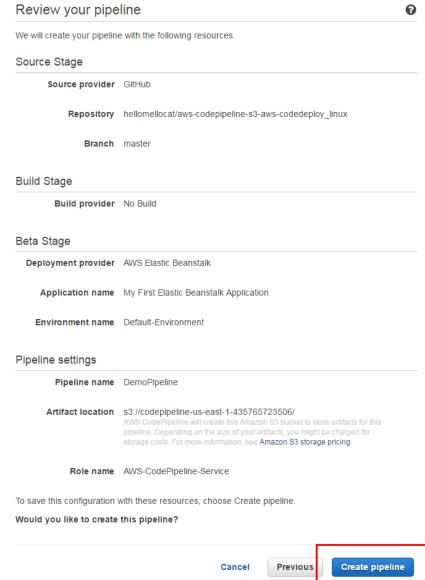
Step 2: Source

Step 3: Build

Step 4: Beta

Step 5: Service Role

Step 6: Review



b. Une fois votre pipeline créé, la page d'état du pipeline s'ouvre et le pipeline commence à s'exécuter automatiquement. Vous pouvez voir l'état d'avancement ainsi que les messages de réussite et d'échec à mesure que le pipeline effectue chaque action.

Pour vérifier que votre pipeline s'est exécuté correctement, surveillez son avancement à chaque étape. L'état de chaque étape passe de *No executions yet* à *In Progress*, puis affiche soit *Succeeded* soit *Failed*. Le pipeline doit terminer la première exécution en quelques minutes.



#### Pipeline created

Congratulations! The pipeline DemoPipeline has been created. Now that you have a pipeline, here are some different ways to start using it.

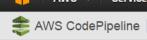
- · Edit your pipeline to add more stages or actions, such as a test or production stages. Learn more
- · Enable or disable transitions between stages to control what stages run automatically in a pipeline. Learn more
- · Manually start a run through your pipeline. Learn more

# DemoPipeline

View progress and manage your pipeline.



c. Dans la zone d'état de l'étape Bêta, cliquez sur AWS Elastic Beanstalk.						





Congratulations! The pipeline DemoPipeline has been created. Now that you have a pipeline, here are some different ways to start using it.

- · Edit your pipeline to add more stages or actions, such as a test or production stages. Learn more
- Enable or disable transitions between stages to control what stages run automatically in a pipeline. Learn more
   Manually start a run through your pipeline. Learn more

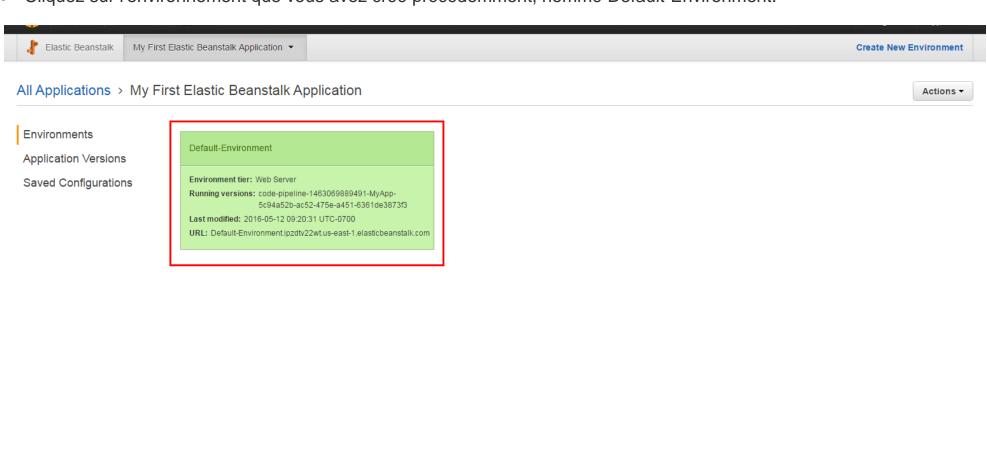
### DemoPipeline

View progress and manage your pipeline.

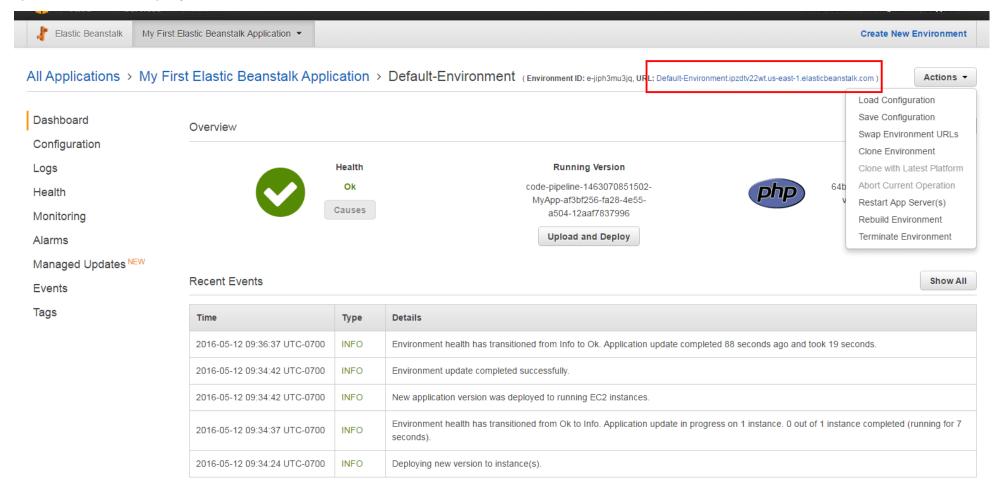


0

- d. La console AWS Elastic Beanstalk s'ouvre et présente les détails du déploiement.
- Cliquez sur l'environnement que vous avez créé précédemment, nommé Default-Environment.



e. Cliquez sur l'URL qui apparaît dans la partie supérieure droite de la page pour afficher l'exemple de site web que vous avez déployé.



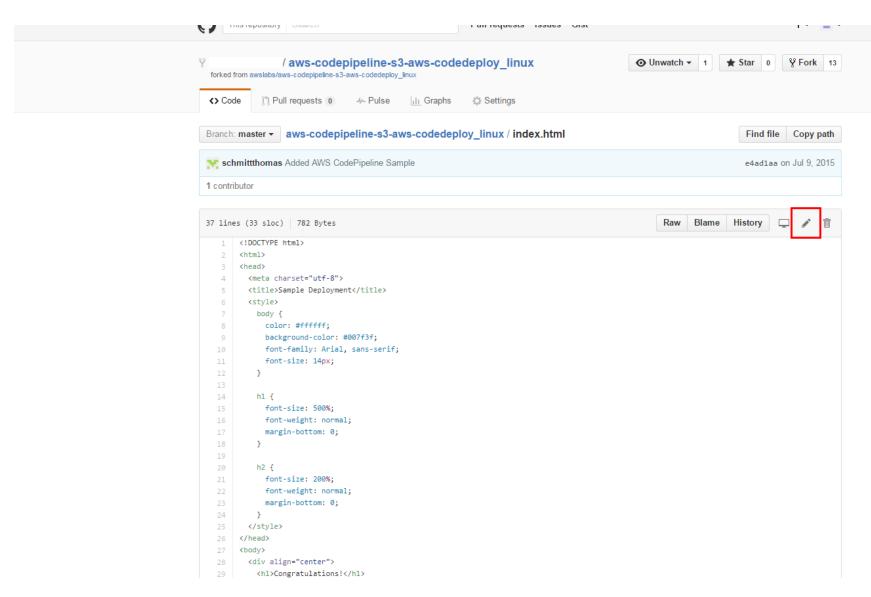
(cliquez pour zoomer)

# Étape 5 : Valider une modification, puis mettre à jour l'application

Au cours de cette étape, vous allez réviser l'exemple de code, puis valider les modifications dans votre référentiel. CodePipeline détecte l'exemple de code mis à jour puis commence automatiquement à le déployer dans votre instance EC2 via Elastic Beanstalk.

Notez que l'exemple de page web que vous avez déployé fait référence à AWS CodeDeploy, un service qui automatise les déploiements de code. Dans CodePipeline, CodeDeploy constitue une alternative à l'utilisation d'Elastic Beanstalk pour les actions de déploiement. Actualisons l'exemple de code afin qu'il indique que vous l'avez déployé avec Elastic Beanstalk.

- a. Examinez la copie du référentiel que vous avez dupliqué dans GitHub.
  - Ouvrez index.html
  - Sélectionnez l'icône Edit.

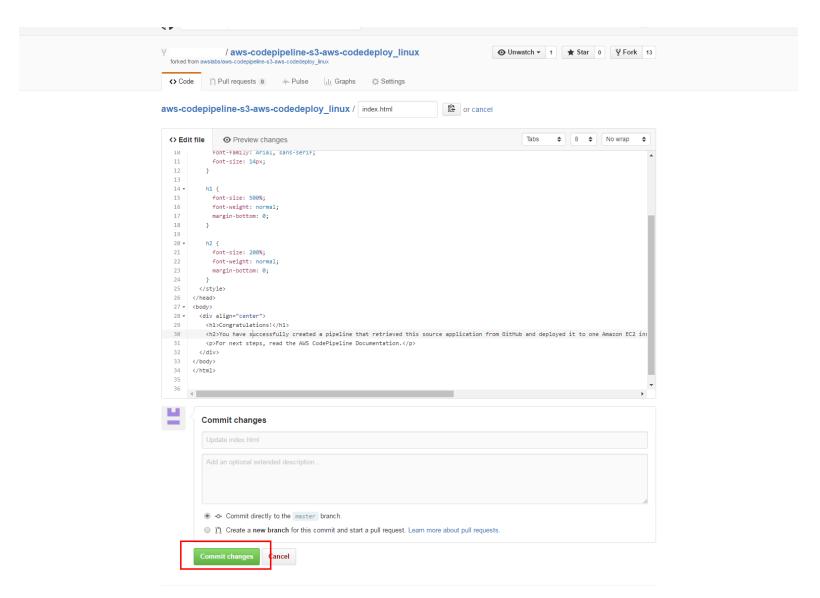


(cliquez pour zoomer)

### b. Mettez à jour la page Web en effectuant un copier-coller du texte suivant à la ligne 30 :

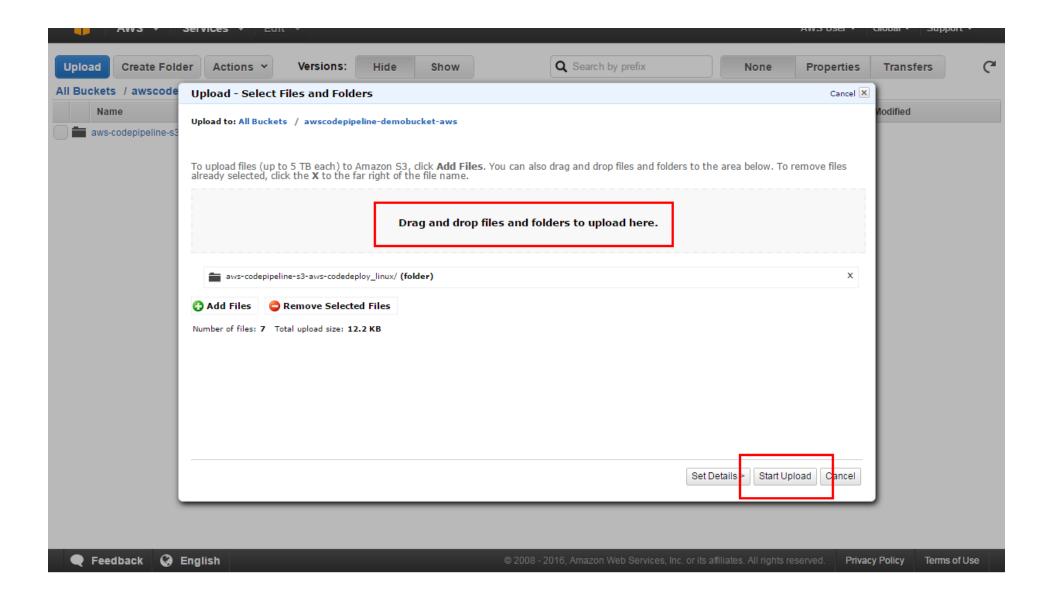
Vous avez créé un pipeline qui a extrait cette application source de GitHub et vous l'avez déployé dans une instance Amaz on EC2 au moyen d'AWS Elastic Beanstalk. Vous vous rapprochez d'une étape de l'adoption du déploiement continu !

```
Tabs
                                                                                                          <> Edit file
              Preview changes
10
           font-family: Arial, sans-serif;
11
           font-size: 14px;
12
         }
13
14 -
         h1 {
15
           font-size: 500%;
16
           font-weight: normal;
           margin-bottom: 0;
17
18
 19
20 -
         h2 {
 21
           font-size: 200%;
 22
           font-weight: normal;
23
           margin-bottom: 0;
24
         }
25
        </style>
26
      </head>
27 🕶
      <body>
       <div align="center">
 28 -
         <h1>Congratulations (</h1>
30
         <h2>You have successfully created a pipeline that retrieved this source application from GitHub and deployed it to one Amazon EC2 in:
         For next steps, read the AWS CodePipeline Documentation.
31
        </div>
32
33
      </body>
34
      </html>
35
 36
```



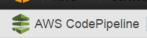
• c. Validez la modification dans le référentiel.

```
untitled
                          index.html
    <!DOCTYPE html>
    <html>
    <head>
      <meta charset="utf-8">
      <title>Sample Deployment</title>
        body {
          color: #ffffff;
          background-color: #007f3f;
          font-family: Arial, sans-serif;
11
          font-size: 14px;
12
        h1 {
          font-size: 500%;
          font-weight: normal;
17
          margin-bottom: 0;
        h2 {
          font-size: 200%;
22
          font-weight: normal;
23
          margin-bottom: 0;
25
      </style>
    </head>
      <div align="center">
        <h1>Congratulations!</h1>
        <h2>You have successfully created a pipeline that retrieved this source application from Amazon S3 and
        deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk. You're one step closer to practicing
        continuous delivery!</h2>
        For next steps, read the AWS CodePipeline Documentation.
32
      </div>
    </body>
    </html>
```



```
untitled
                          index.html
    <!DOCTYPE html>
    <html>
    <head>
      <meta charset="utf-8">
      <title>Sample Deployment</title>
        body {
          color: #ffffff;
          background-color: #007f3f;
          font-family: Arial, sans-serif;
11
          font-size: 14px;
12
        h1 {
          font-size: 500%;
          font-weight: normal;
17
          margin-bottom: 0;
        h2 {
          font-size: 200%;
22
          font-weight: normal;
23
          margin-bottom: 0;
25
      </style>
    </head>
      <div align="center">
        <h1>Congratulations!</h1>
        <h2>You have successfully created a pipeline that retrieved this source application from Amazon S3 and
        deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk. You're one step closer to practicing
        continuous delivery!</h2>
        For next steps, read the AWS CodePipeline Documentation.
32
      </div>
    </body>
    </html>
```

d. Retournez au pipeline dans la console CodePipeline. Dans quelques minutes, vous verrez la Source devenir bleue, ce qui indique que le pipeline a détecté les modifications que vous avez apportées à votre référentiel source. Le code mise à jour est ensuite transféré automatiquement vers Elastic Beanstalk.
<ul> <li>Une fois que l'état du pipeline affiche Succeeded, dans la zone d'état de l'étape Beta, cliquez sur AWS Elastic Beanstalk.</li> </ul>





Congratulations! The pipeline DemoPipeline has been created. Now that you have a pipeline, here are some different ways to start using it.

- · Edit your pipeline to add more stages or actions, such as a test or production stages. Learn more
- · Enable or disable transitions between stages to control what stages run automatically in a pipeline. Learn more
- Manually start a run through your pipeline. Learn more

### DemoPipeline

View progress and manage your pipeline.



×

0

- e. La console AWS Elastic Beanstalk s'ouvre et présente les détails du déploiement. Sélectionnez l'environnement que vous avez créé précédemment, nommé Default-Environment.
- f. Cliquez sur l'URL qui apparaît dans la partie supérieure droite de la page pour afficher à nouveau l'exemple de site web que vous avez déployé. Votre texte a été mis à jour automatiquement via le pipeline de déploiement continu.

#### All Applications > My First Elastic Beanstalk Application

Environments

**Application Versions** 

Saved Configurations

Default-Environment

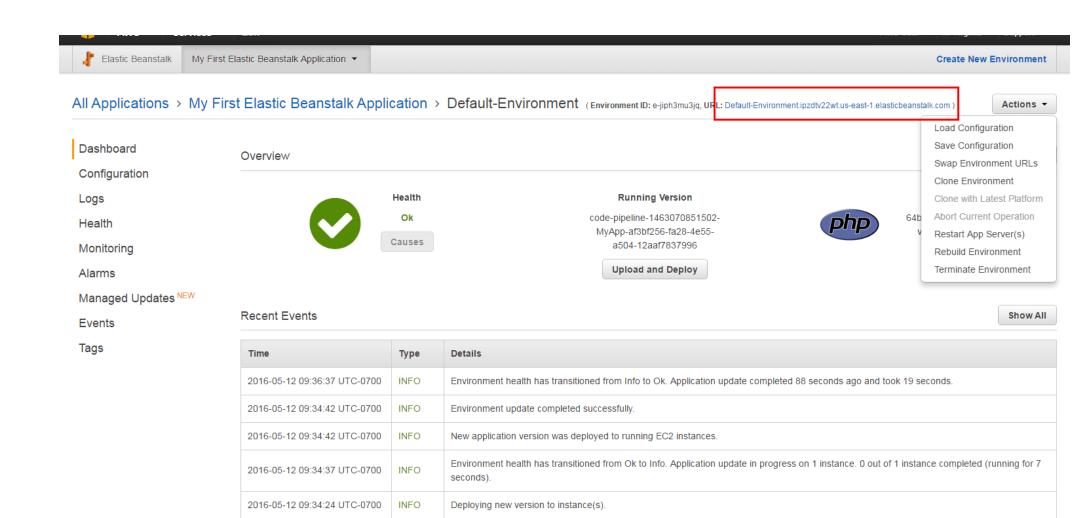
Environment tier: Web Server

Running versions: code-pipeline-1463069889491-MyApp-

5c94a52b-ac52-475e-a451-6361de3873f3

Last modified: 2016-05-12 09:20:31 UTC-0700

URL: Default-Environment.ipzdtv22wt.us-east-1.elasticbeanstalk.com

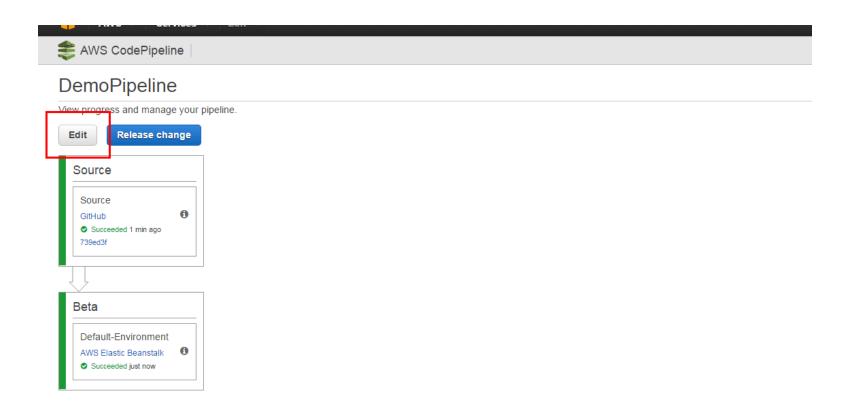


Cliquez à présént action dans l'environnement de dev et cliquez sur le menu 'swap environment url 'afin d'intervertir la dev et la prod

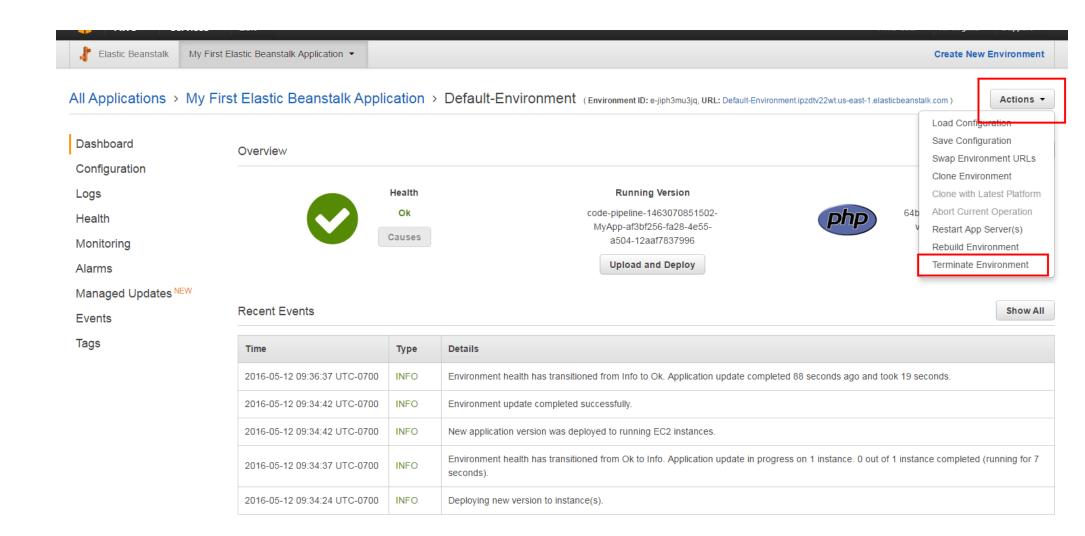
## Étape 6 : Nettoyer vos ressources

Afin d'éviter des frais à l'avenir, vous allez supprimer toutes les ressources que vous avez lancées au cours de ce didacticiel, à savoir le pipeline, l'application Elastic Beanstalk et la source que vous avez configurée pour héberger le code.

- a. Vous devez d'abord supprimer votre pipeline :
- Dans la vue du pipeline, cliquez sur Edit.
- Cliquez sur Delete.
- Saisissez le nom de votre pipeline, puis cliquez sur Delete.



- b. Vous devez ensuite supprimer votre application Elastic Beanstalk :
- Accédez à la console Elastic Beanstalk.
- Cliquez sur Actions.
- Cliquez ensuite sur Terminate Environment.



- c. Si vous avez créé un compartiment S3 pour ce didacticiel, supprimez-le :
  - Accédez à la console S3.

- Cliquez avec le bouton droit sur le nom du compartiment, puis sélectionnez Delete Bucket.
- Lorsqu'un message de confirmation apparaît, saisissez le nom du compartiment, puis cliquez sur Delete.

