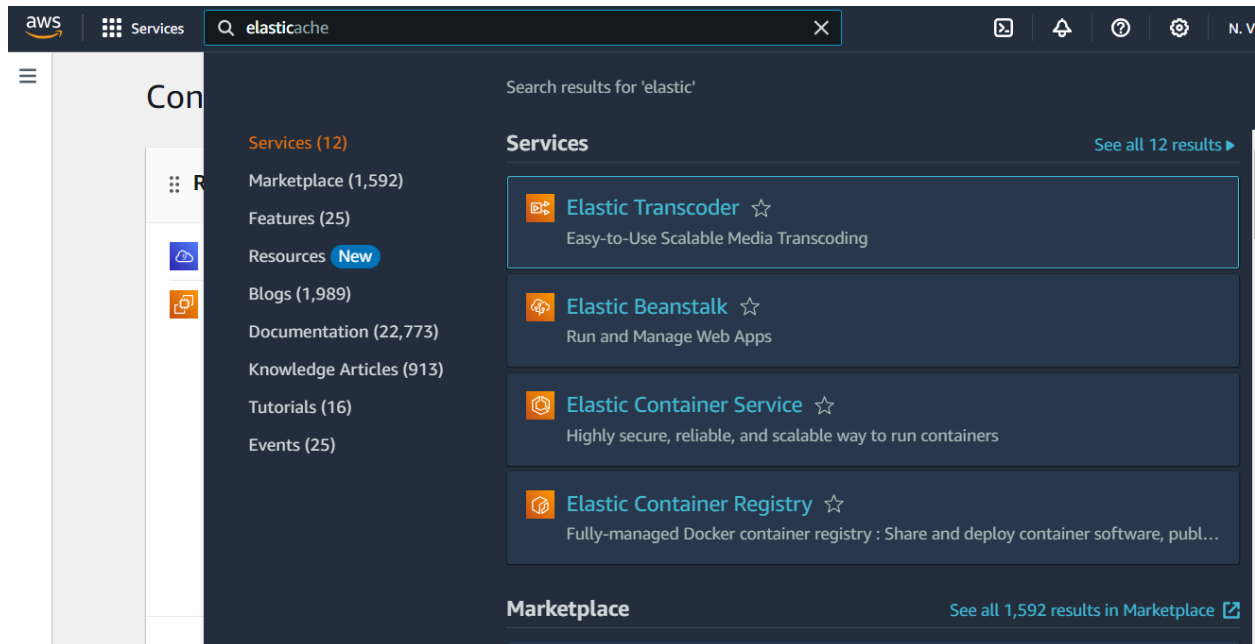
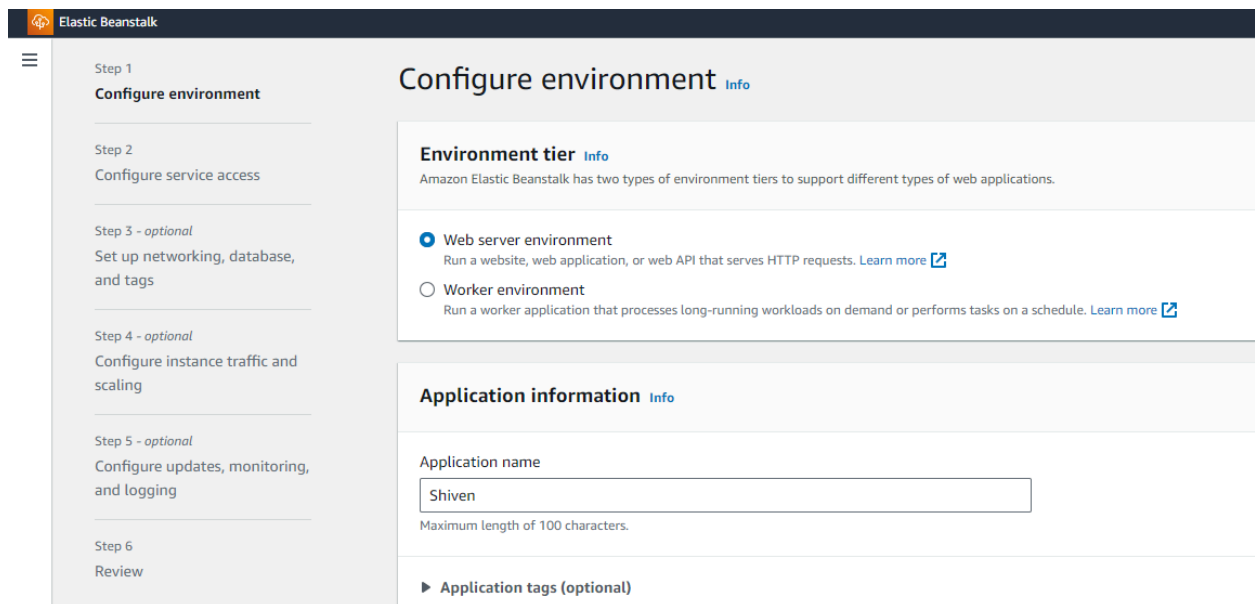


Step 1: Login to your AWS console. Search for Elastic Beanstalk in the searchbar near services



Step 2: Go to Elastic Beanstalk and click on Create Application



Name: Shiven Bansal

Roll no : 03

Class: D15C

Step 3: Enter the name of your application. Scroll down and in the platform, select platform as PHP. Keep the application code as Sample Application. Set the instance to single instance. Click on NEXT.

Environment name

Shiven-env

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

Leave blank for autogenerated value

.us-east-1.elasticbeanstalk.com

Check availability

Environment description

Platform [Info](#)

Platform type

☒ Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Step 4: Use an existing service role and choose whatever service role is available on your account.

The screenshot shows the 'Configure service access' step in the AWS IAM console. The left sidebar lists steps 2 through 6, with step 2 being the current step. The main content area is titled 'Service access' and contains the following sections:

- Service role:** Two radio buttons are present: 'Create and use new service role' (unselected) and 'Use an existing service role' (selected).
- Existing service roles:** A dropdown menu shows 'AWSCloud9SSMAccessRole'.
- EC2 key pair:** A dropdown menu shows 'Choose a key pair'.
- EC2 instance profile:** A dropdown menu shows 'AWScloud9SSMInstanceProfile'.

At the bottom of the main content area, there is a 'View permission details' button. At the bottom of the entire console window, there are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next'.

Step 5: Review the settings that you have set up for your application and submit your application.

The screenshot shows the 'Review' step in the AWS IAM console. The left sidebar lists steps 1 through 6, with step 1 being the current step. The main content area is titled 'Review' and contains the following sections:

- Step 1: Configure environment:** A table displays environment information.
- Step 2: Configure service access:** A section for service access configuration.

Environment information	
Environment tier	Application name
Web server environment	Shiven
Environment name	Application code
Shiven-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	

Step 6: Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

<https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>

The screenshot shows a GitHub repository page for 'aws-codepipeline-s3-codedeploy-linux-2.0' by user 'shivenbansal12'. The repository is a fork of 'imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0'. The page displays the repository name, a search bar, and navigation tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the repository name, it shows '1 Branch' and '0 Tags'. A message indicates the branch is up to date with the upstream repository. A list of commits is shown, including updates to README.md, .github, dist, scripts, CODE_OF_CONDUCT.md, CONTRIBUTING.md, and LICENSE.

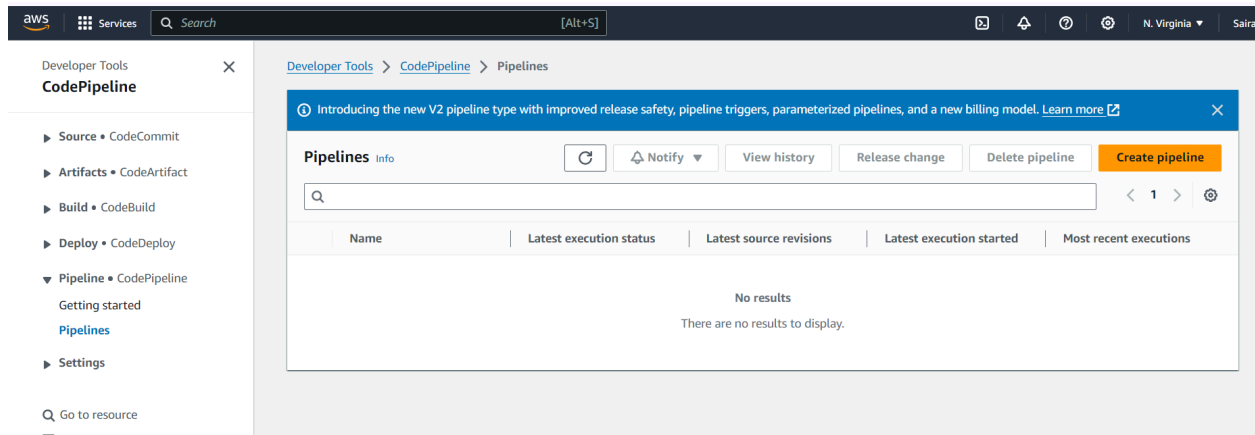
Commit	Message	Time
8fd5da5	Update README.md	3 years ago
	Adding template	7 years ago
	Added dist folder	9 years ago
	s3 setup and s3 set cahce control scripts	3 years ago
	Adding CONTRIBUTING/CoC	7 years ago
	Adding CONTRIBUTING/CoC	7 years ago
	Added AWS CodePipeline Sample	9 years ago
	Update README.md	3 years ago

Step 7: Search CodePipeline in the services tab and click on it.

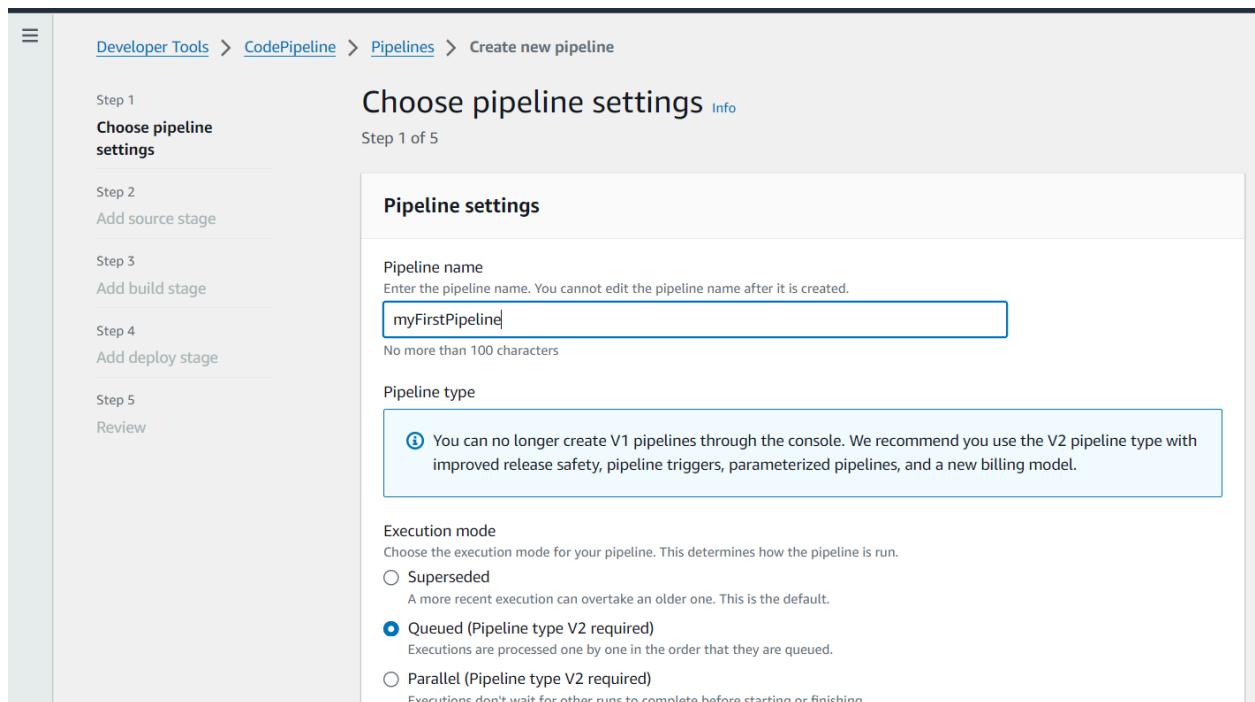
The screenshot shows the AWS IAM console search results for 'codepipeline'. The search bar at the top contains 'codepipeline'. The results are displayed under the 'Services' tab, showing a list of services including CodePipeline, Amazon Q Developer (Including Amazon CodeWhisperer), CodeCommit, and AWS Signer. The 'CodePipeline' service is highlighted, with the description 'Release Software using Continuous Delivery'.

Service	Description
CodePipeline	Release Software using Continuous Delivery
Amazon Q Developer (Including Amazon CodeWhisperer)	Build applications faster, and spend less time solving software development problems.
CodeCommit	Store Code in Private Git Repositories
AWS Signer	

Step 8: Click on Create Pipeline.



Step 9: Give a name to your Pipeline. A new service role would be created with the name of the Pipeline



Step 10: Select a source provider (as Github (Version 2)). Click on Connect to Github to connect your github.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

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.

GitHub (Version 2) ▼

 **New GitHub version 2 (app-based) action**
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

or

Repository name

Step 11: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or install a new app. If this is your first time, click on Install a new app.

aws | Services | Search | More ▼

Developer Tools > ... > Create connection

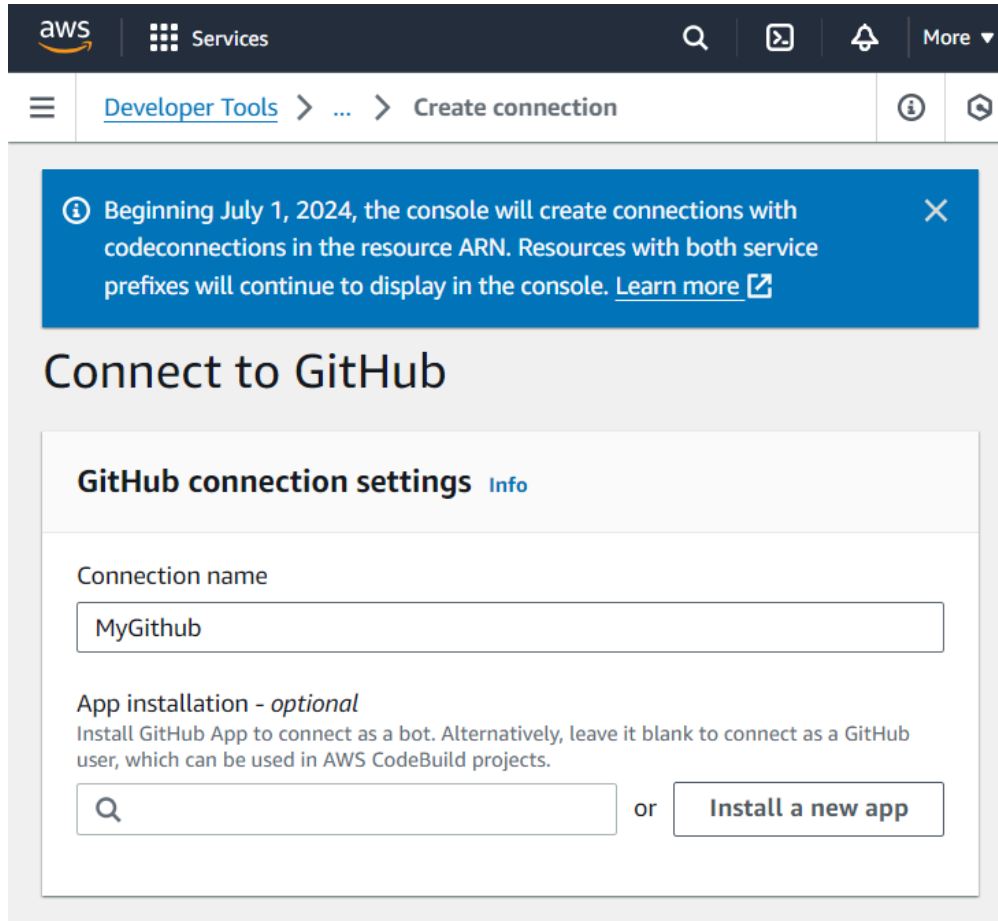
Create a connection Info

Create GitHub App connection Info

Connection name

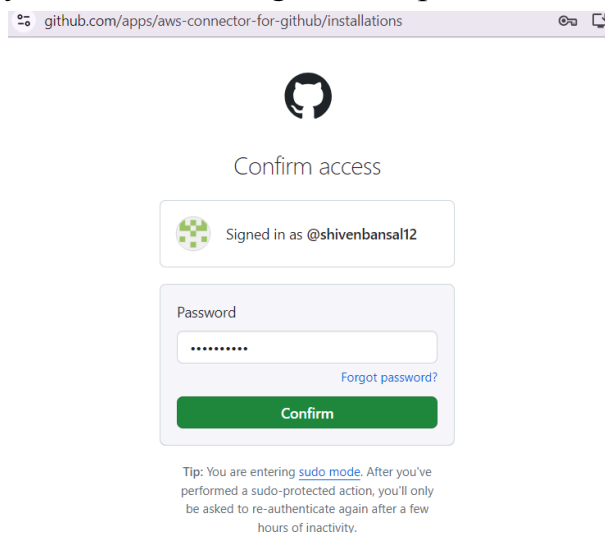
MyGithub

► Tags - optional



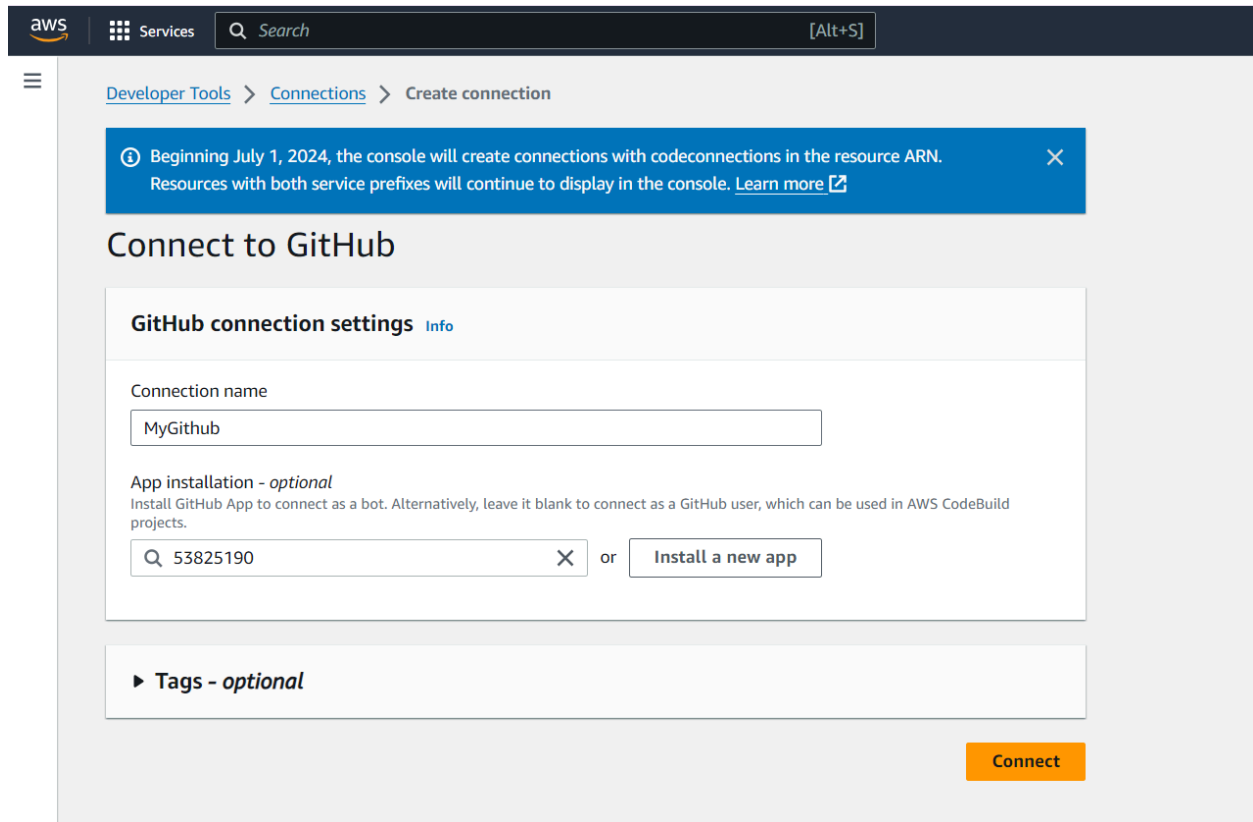
The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services' menu, search bar, and notification bell. Below this, a breadcrumb trail shows 'Developer Tools' > 'Create connection'. A blue informational banner at the top states: 'Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. [Learn more](#)'. The main heading is 'Connect to GitHub'. Underneath, the 'GitHub connection settings' section includes a text input for 'Connection name' with the value 'MyGithub'. Below that, the 'App installation - optional' section provides instructions: 'Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.' It features a search input field and an 'Install a new app' button.

Step 12: This will direct you to install AWS Connector on your GitHub. Install it to your account and give it its permissions



The screenshot shows the GitHub 'Confirm access' page for the 'aws-connector-for-github' app. The browser address bar shows 'github.com/apps/aws-connector-for-github/installations'. The page features the GitHub logo and the title 'Confirm access'. A box indicates the user is 'Signed in as @shivenbansal12'. Below this is a 'Password' field with masked characters and a 'Forgot password?' link. A green 'Confirm' button is positioned below the password field. At the bottom, a tip states: 'Tip: You are entering [sudo mode](#). After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.'

Step 13: After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use.



aws Services Search [Alt+S]

Developer Tools > Connections > Create connection

Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. [Learn more](#)

Connect to GitHub

GitHub connection settings [Info](#)

Connection name

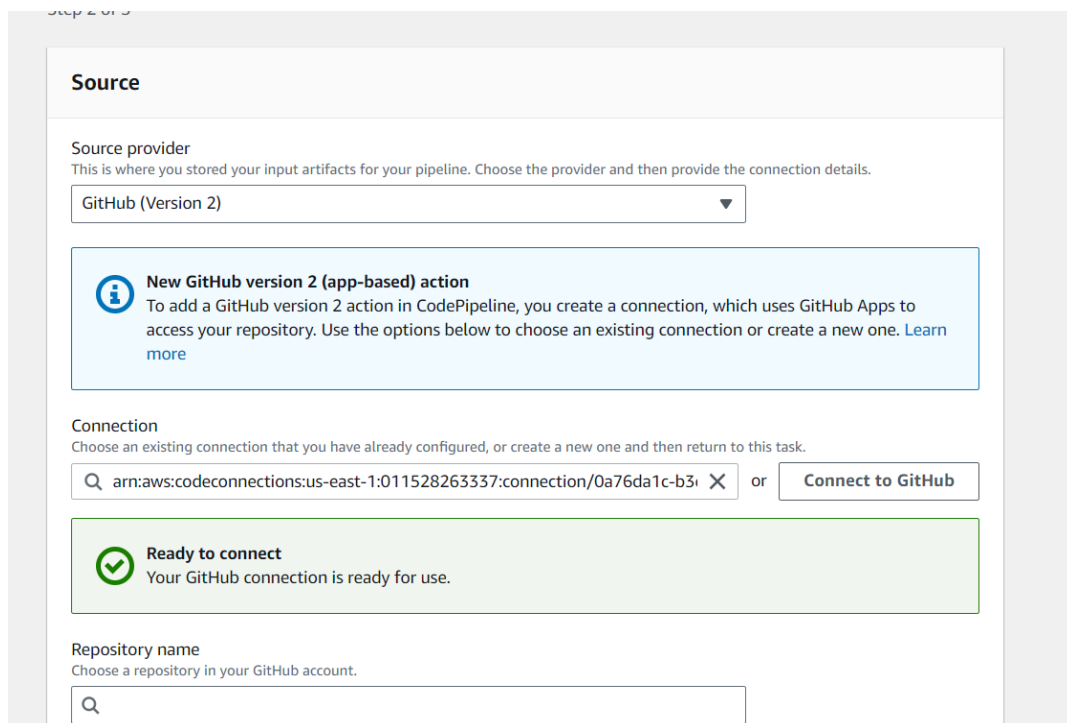
MyGithub

App installation - *optional*
Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.

53825190 or [Install a new app](#)

Tags - *optional*

[Connect](#)



Step 2 of 3

Source

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

GitHub (Version 2)

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codeconnections:us-east-1:011528263337:connection/0a76da1c-b3... or [Connect to GitHub](#)

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

Step 14: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

The screenshot shows the AWS CodePipeline console interface. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and a keyboard shortcut '[Alt+S]'. On the left, a sidebar shows a 'Review' button. The main content area is titled 'New GitHub version 2 (app-based) action'. It includes a blue information box explaining that a connection must be created. Below this, the 'Connection' section shows a dropdown menu with the value 'arn:aws:codeconnections:us-east-1:011528263337:connection/0a76da1c-b3...' and a 'Connect to GitHub' button. A green 'Ready to connect' box indicates the connection is ready. The 'Repository name' section has a dropdown menu showing 'shivenbansal12/aws-codepipeline-s3-codedeploy-linux-2.0'. The 'Default branch' section has a dropdown menu showing 'master'. The 'Output artifact format' section has two options: 'CodePipeline default' (selected) and 'Full clone'. A blue information box at the bottom states: 'You can add additional sources and triggers by editing the pipeline after it is created.'

Step 15: Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.

The screenshot shows the AWS CodePipeline console interface, specifically the 'Trigger' configuration section. The 'Output artifact format' section is visible at the top, with 'CodePipeline default' selected. The 'Trigger' section has a title 'Trigger' and a sub-section 'Trigger type' with the instruction 'Choose the trigger type that starts your pipeline.' There are three radio button options: 'No filter' (selected), 'Specify filter', and 'Do not detect changes'. The 'No filter' option has a description: 'Starts your pipeline on any push and clones the HEAD.' The 'Specify filter' option has a description: 'Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.' The 'Do not detect changes' option has a description: 'Don't automatically trigger the pipeline.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

Step 16: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.

The screenshot shows the AWS CodePipeline console. The breadcrumb navigation is **Developer Tools > CodePipeline > Pipelines > Create new pipeline**. On the left, a sidebar lists the steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled **Add build stage** (Step 3 of 5). It contains a section for **Build - optional** with a **Build provider** dropdown menu. At the bottom right, there are buttons for **Cancel**, **Previous**, **Skip build stage**, and **Next**.

The screenshot shows the AWS CodePipeline console with the **Deploy** step configuration. The breadcrumb navigation is **Developer Tools > CodePipeline > Pipelines > Create new pipeline**. The sidebar shows Step 4 (Add deploy stage), Step 5, and Review. The main area is titled **Deploy**. It contains the following fields: **Deploy provider** (AWS Elastic Beanstalk), **Region** (US East (N. Virginia)), **Input artifacts** (SourceArtifact), **Application name** (MyApp), and **Environment name** (MyApp-env). There is a checkbox for **Configure automatic rollback on stage failure**. At the bottom right, there are buttons for **Cancel**, **Previous**, and **Next**.

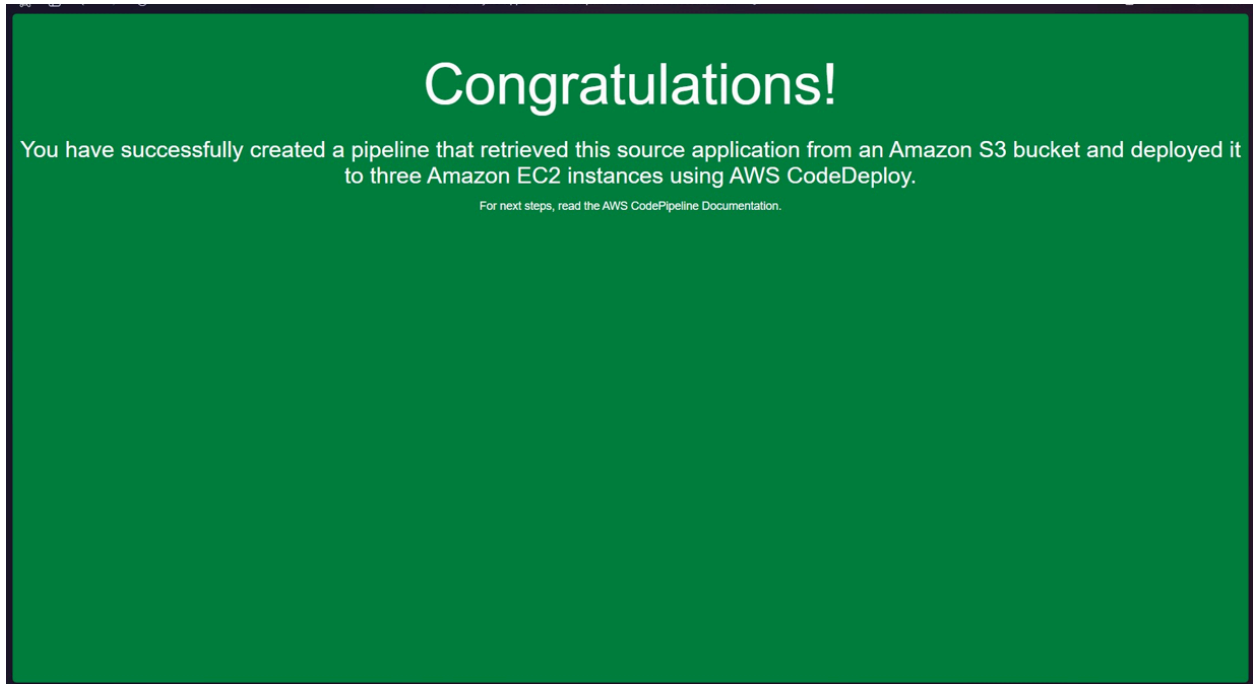
Step 17: Check all the information and click on create Pipeline.

The screenshot shows the AWS CodePipeline console interface. At the top, there's a header with the AWS logo, 'Services', a search bar, and a '[Alt+S]' shortcut. Below the header, a sidebar on the left contains a menu with 'Build action provider', 'Build stage', and 'No build'. The main content area is titled 'Step 4: Add deploy stage'. It contains a form with the following fields: 'Deploy action provider' (set to 'AWS Elastic Beanstalk'), 'ApplicationName' (set to 'MyApp'), 'EnvironmentName' (set to 'MyApp-env'), and 'Configure automatic rollback on stage failure' (set to 'Disabled'). At the bottom right of the form, there are three buttons: 'Cancel', 'Previous', and 'Create pipeline' (highlighted in orange).

Step 18: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy. Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.

The screenshot shows the AWS CodePipeline console interface after a pipeline has been created. A green banner at the top reads 'Success Congratulations! The pipeline myFirstPipeline has been created.' with a button to 'Create a notification rule for this pipeline'. Below the banner, the breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > myFirstPipeline'. The main content area is titled 'myFirstPipeline' and shows 'Pipeline type: V2' and 'Execution mode: QUEUED'. There are buttons for 'Notify', 'Edit', 'Stop execution', 'Clone pipeline', and 'Release change'. A section titled 'Source' shows a 'Succeeded' status with a 'Just now' timestamp. Below this, there's a box containing the source configuration: 'Source', 'GitHub (Version 2)', 'Succeeded - Just now', and a 'View details' button. At the bottom, there's a link to '8fd5da54' and a note 'Source: Update README.md'.

Step 19: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain. And then this will be shown



Step 20: Now, we make some changes to the index.html file in the github. For eg: If you make some changes to the <h2> tag.

Once the changes are committed, when the website is refreshed, the changes can be seen.

