

Project 2

Create the (continuous delivery) cd pipeline

With help of AWS Service [AWS Elastic Beanstalk](#), [AWS CodeBuild](#), and [AWS CodePipeline](#)

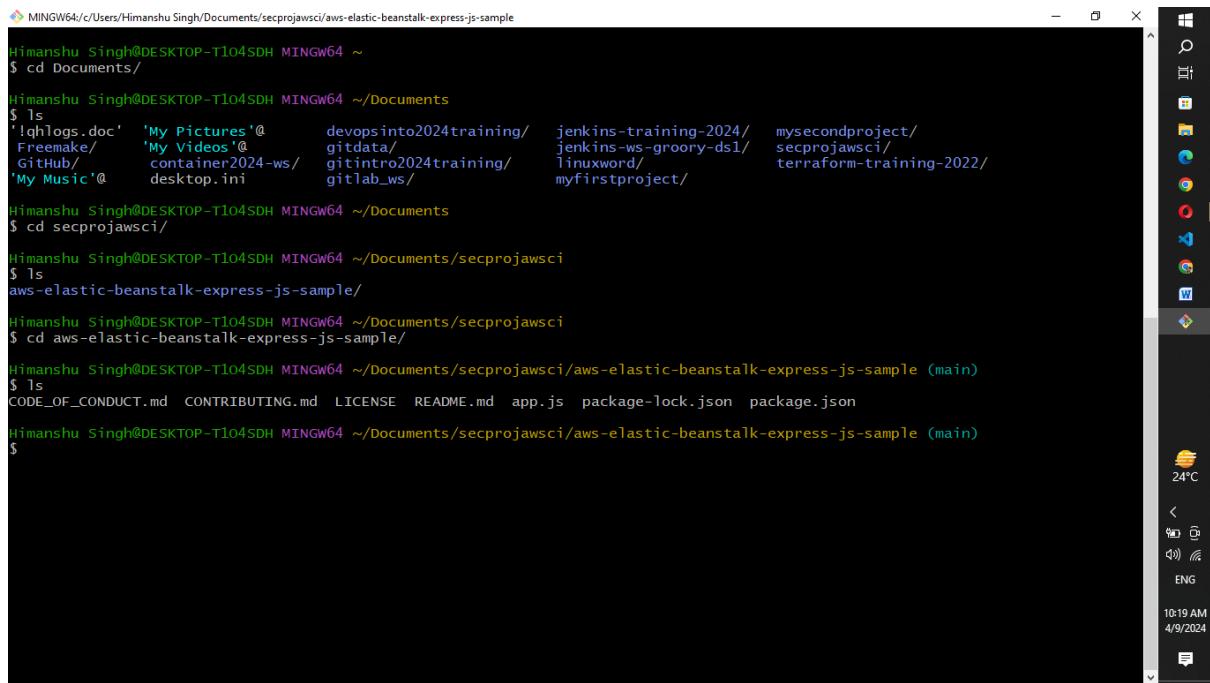
Step 1 - Made Setup for the github repository.

The screenshot shows a GitHub repository page for 'aws-elastic-beanstalk-express-js-sample'. The repository is public and was forked from 'aws-samples/aws-elastic-beanstalk-express-js-sample'. The main branch has 1 commit ahead of the upstream. The repository contains files like .github, .gitignore, CODE_OF_CONDUCT.md, CONTRIBUTING.md, LICENSE, README.md, and aod.is. The repository has 5 commits, 0 stars, 0 forks, and 0 releases. The page includes a sidebar with weather information (24°C), system status, and a timestamp (10:08 AM 4/9/2024).

This screenshot is identical to the one above, but a 'Clone' context menu is open over the repository name. It provides options for cloning via HTTPS, SSH, or GitHub CLI, and includes a 'Copy url to clipboard' button. The clipboard URL is https://github.com/HimanshuSingh1706/aws-elast... The rest of the repository details and sidebar are visible.

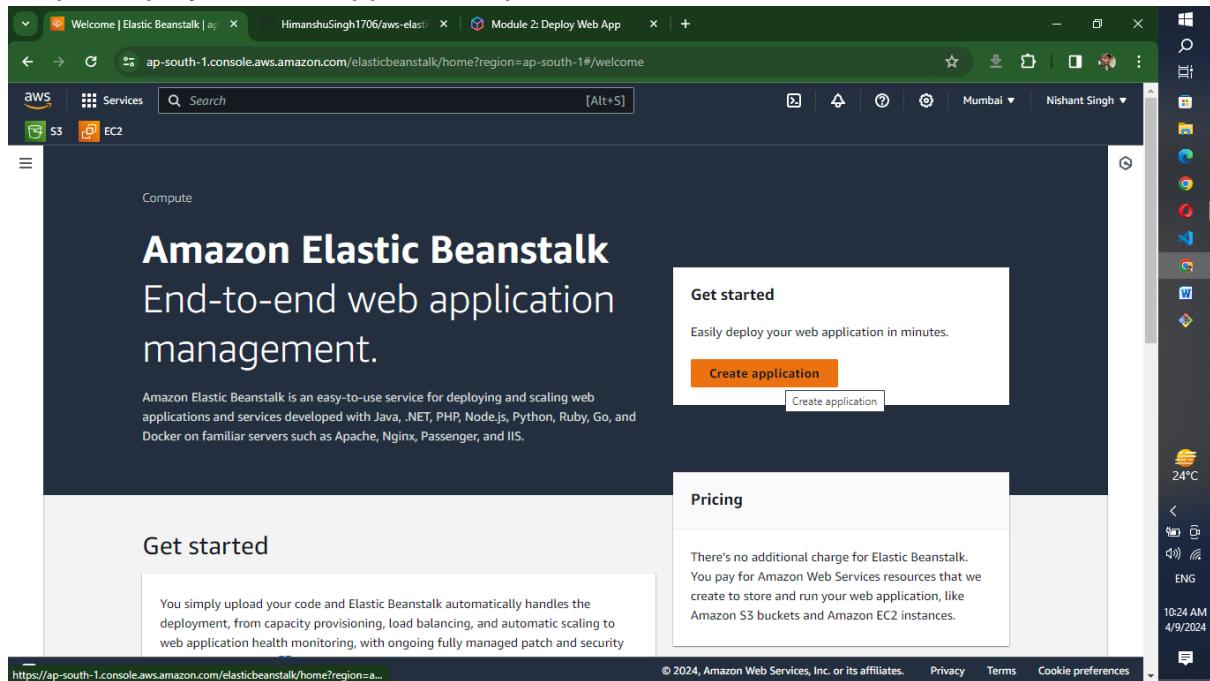
Clone file on gitbash with help of #git clone command

#Git clone - <https://github.com/HimanshuSingh1706/aws-elastic-beanstalk-express-js-sample.git>



```
MINGW64:/c/Users/Himanshu Singh/Documents/secprojawscli/aws-elastic-beanstalk-express-js-sample
Himanshu Singh@DESKTOP-T1O4SDH MINGW64 ~
$ cd Documents/
Himanshu Singh@DESKTOP-T1O4SDH MINGW64 ~/Documents
$ ls
'!qhlogs.doc'  'My Pictures'@      devopsinto2024training/  jenkins-training-2024/   mysecondproject/
'Freemake/'     'My Videos'@       gitdata/             jenkins-ws-groovy-dsl/   secprojawscli/
'GitHub/'       container2024-ws/  gitintro2024training/  linuxword/           terraform-training-2022/
'My Music'@     desktop.ini       gitlab_ws/          myFirstProject/
Himanshu Singh@DESKTOP-T1O4SDH MINGW64 ~/Documents
$ cd secprojawscli/
Himanshu Singh@DESKTOP-T1O4SDH MINGW64 ~/Documents/secprojawscli
$ ls
aws-elastic-beanstalk-express-js-sample/
Himanshu Singh@DESKTOP-T1O4SDH MINGW64 ~/Documents/secprojawscli
$ cd aws-elastic-beanstalk-express-js-sample/
Himanshu Singh@DESKTOP-T1O4SDH MINGW64 ~/Documents/secprojawscli/aws-elastic-beanstalk-express-js-sample (main)
$ ls
CODE_OF_CONDUCT.md CONTRIBUTING.md LICENSE README.md app.js package-lock.json package.json
Himanshu Singh@DESKTOP-T1O4SDH MINGW64 ~/Documents/secprojawscli/aws-elastic-beanstalk-express-js-sample (main)
$
```

Step 2 deploy the web app with help of AWS Elastic Beanstalk



Welcome | Elastic Beanstalk | ap-south-1.console.aws.amazon.com | Module 2: Deploy Web App | X | Module 2: Deploy Web App | X | +

ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/welcome

AWS Services Search [Alt+S] Mumbai Nishant Singh

Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Create application

Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=a... © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'Configure environment' step in the AWS Elastic Beanstalk console. The left sidebar lists steps 1 through 6. Step 1 is selected, titled 'Configure environment'. The main content area is titled 'Configure environment' and contains two sections: 'Environment tier' and 'Application information'.
Environment tier
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.
 Web server environment
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
 Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)
Application information
Application name: firstwebapp
Maximum length of 100 characters.
▼ Application tags (optional)
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

The screenshot shows the 'Configure environment' step in the AWS Elastic Beanstalk console. The left sidebar lists steps 1 through 6. Step 1 is selected, titled 'Configure environment'. The main content area is titled 'Configure environment' and contains two sections: 'Platform' and 'Application code'.
Platform
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: Node.js
Platform branch: Node.js 20 running on 64bit Amazon Linux 2023
Platform version: 6.1.2 (Recommended)
Application code
 Sample application

Screenshot of the AWS Elastic Beanstalk "Configure service access" step.

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role:

Use an existing service role
 Create and use new service role

Existing service roles:

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

EC2 key pair:

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

EC2 instance profile:

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

Review

Step 1: Configure environment

Environment tier	Application name
Web server environment	firstwebapp

Step 2: Configure service access

Service role	EC2 key pair	EC2 instance profile
EC-elasticbeanstalk-role	first_project_key	EC-elasticbeanstalk-role

[View permission details](#)

Screenshot of the AWS Elastic Beanstalk console showing the configuration of a new environment.

Configure environment - review

Step 5: optional
[Configure updates, monitoring, and logging](#)

Step 6 Review

Step 2: Configure service access

Service access [Info](#)
Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role arn:aws:iam:975050016757:role/EC- elascticbeanstalk-role	EC2 key pair first_project_key	EC2 instance profile EC-elasicbeanstalk-role
---	-----------------------------------	---

Step 3: Set up networking, database, and tags

Networking, database, and tags [Info](#)
Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

No options configured

Tags

Elastic Beanstalk is launching your environment. This will take a few minutes.

Firstwebapp-env [Info](#)

Environment overview

Health Unknown	Environment ID e-hz4pseive
Domain -	Application name firstwebapp

Platform [Change version](#)

Platform Node.js 20 running on 64bit Amazon Linux 2023/6.1.2
Running version -
Platform state Supported

Events [Events \(2\)](#) [Info](#)

Filter events by text, property or value

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

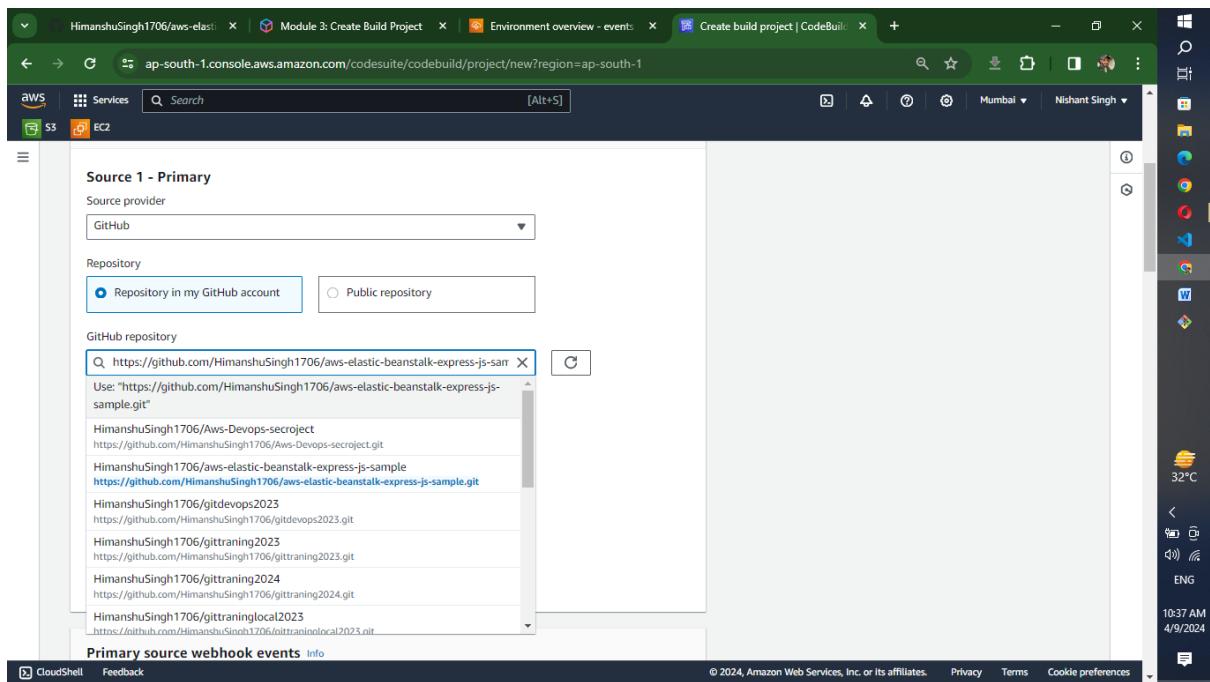
CloudShell Feedback

CloudShell Feedback

The screenshot shows the AWS Elastic Beanstalk environment overview for 'Firstwebapp-env'. A green success message at the top says 'Environment successfully launched.' The main panel displays the environment's configuration, including its Health status (Warning), Environment ID (e-hz4pseive), Domain (Firstwebapp-env.eba-idff7pmd.ap-south-1.elasticbeanstalk.com), Application name (firstwebapp), Platform (Node.js 20 running on 64bit Amazon Linux 2023/6.1.2), and Platform state (Supported). Below this, tabs for Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags are visible. The 'Events' tab shows 11 events. The bottom of the screen includes standard AWS navigation and footer links.

Step 3 create build project – AWS codebuild

The screenshot shows the 'Create build project' wizard in the AWS CodeBuild console. The first step, 'Project configuration', is completed with a project name 'firstwebapp'. The second step, 'Additional configuration', is partially filled with 'GitHub' selected as the source provider. The third step, 'Source', is shown with 'Source 1 - Primary' configured to use a GitHub repository. The bottom of the screen includes standard AWS navigation and footer links.



Environment overview - events

Create build project | CodeBuild

Environment

Provisioning model Info

- On-demand
Automatically provision build infrastructure in response to new builds.
- Reserved capacity
Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

Environment image

- Managed image
Use an image managed by AWS CodeBuild
- Custom image
Specify a Docker image

Compute

- EC2
Optimized for flexibility during action runs
- Lambda
Optimized for speed and minimizes the start up time of workflow actions

Operating system

Amazon Linux

Create build project | CodeBuild

Operating system

Amazon Linux

Runtime(s)

Standard

Image

aws/codebuild/amazonlinux2-x86_64-standard:5.0

Image version

Always use the latest image for this runtime version

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

codebuild-firstwebapp-service-role

Type your service role name

Report auto-discover Info

Enable this flag to search build files for supported report file types and generate reports

Auto-discover directory - optional

**/*

The screenshot shows two separate sessions of the AWS CodeBuild console.

Top Session: A user is creating a new build project. In the "Buildspec" section, the "Insert build commands" option is selected, displaying a YAML snippet:

```
version: 0.2
phases:
  build:
    commands:
      - echo "Hello World"
```

Bottom Session: A user has successfully created a project named "firstwebapp". The "Configuration" tab is visible, showing the following details:

Source provider	Primary repository	Artifacts upload location	Service role
GitHub	HimanshuSingh1706/aws-elastic-beanstalk-express-js-sample	-	arn:awsiam::975050016757:role/service-role/codebuild-firstwebapp-service-role
Public builds	Disabled		

The "Build history" tab shows one recent build, which is currently running. Other tabs include "Batch history", "Project details", "Build triggers", and "Metrics".

Screenshot of the AWS CloudWatch Logs showing the build log for the 'firstwebapp' project. The log output is as follows:

```

1 beanstalk-express.js sample
11 [Container] 2024/04/09 05:10:23.955885 Unable to initialize cache download: no paths specified to be cached
12 [Container] 2024/04/09 05:10:24.029341 Configuring ssm agent with target id: codebuild:3d156206-c56b-49ef-8e5a-5ad5072e9710
13 [Container] 2024/04/09 05:10:24.049717 Successfully updated ssm agent configuration
14 [Container] 2024/04/09 05:10:24.050017 Registering with agent
15 [Container] 2024/04/09 05:10:24.087200 Phase found in YAML: 1
16 [Container] 2024/04/09 05:10:24.087216 BUILD: 1 commands
17 [Container] 2024/04/09 05:10:24.087470 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
18 [Container] 2024/04/09 05:10:24.087470 Phase context status code: Message:
19 [Container] 2024/04/09 05:10:24.154701 Entering phase INSTALL
20 [Container] 2024/04/09 05:10:24.176503 Phase complete: INSTALL State: SUCCEEDED
21 [Container] 2024/04/09 05:10:24.176519 Phase context status code: Message:
22 [Container] 2024/04/09 05:10:24.207377 Entering phase PRE_BUILD
23 [Container] 2024/04/09 05:10:24.209000 Phase complete: PRE_BUILD State: SUCCEEDED
24 [Container] 2024/04/09 05:10:24.209014 Phase context status code: Message:
25 [Container] 2024/04/09 05:10:24.240462 Entering phase BUILD
26 [Container] 2024/04/09 05:10:24.240886 Running command
27
28 [Container] 2024/04/09 05:10:24.247210 Phase complete: BUILD State: SUCCEEDED
29 [Container] 2024/04/09 05:10:24.247223 Phase context status code: Message:
30 [Container] 2024/04/09 05:10:24.277973 Entering phase POST_BUILD
31 [Container] 2024/04/09 05:10:24.279544 Phase complete: POST_BUILD State: SUCCEEDED
32 [Container] 2024/04/09 05:10:24.279555 Phase context status code: Message:
33 [Container] 2024/04/09 05:10:24.323614 Set report auto-discover timeout to 5 seconds
34 [Container] 2024/04/09 05:10:24.326615 Expanding base directory path: .
35 [Container] 2024/04/09 05:10:24.326615 Assembling file list
36 [Container] 2024/04/09 05:10:24.326630 Expanding .
37 [Container] 2024/04/09 05:10:24.329744 Expanding file paths for base directory .
38 [Container] 2024/04/09 05:10:24.329758 Assembling file list
39 [Container] 2024/04/09 05:10:24.329762 Expanding */
40 [Container] 2024/04/09 05:10:24.333370 Found 2 file(s)
41 [Container] 2024/04/09 05:10:24.333465 Report auto-discover file discovery took 0.000926 seconds
42 [Container] 2024/04/09 05:10:24.333983 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED
43 [Container] 2024/04/09 05:10:24.333995 Phase context status code: Message:
44

```

Step 4 setup the CD pipeline

The screenshot displays two browser windows side-by-side, both showing the AWS CodePipeline interface.

Top Window (aws.amazon.com/codepipeline/):

- Header:** Shows tabs for 'HimanshuSingh1706/aws', 'Module 4: Create Delivery', 'Environment overview', 'firstwebapp3d156206-c5...', and 'CL/CD Pipeline - AWS CodePipeline'. The URL is 'aws.amazon.com/codepipeline/'.
- Navigation:** AWS logo, Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, Sign In to the Console.
- Content:** AWS CodePipeline Overview, Free AWS Training, and developer tools links. A callout box highlights 'One free active pipeline per month with the AWS Free Tier'.
- Bottom:** Buttons for 'Get started with CodePipeline' and four main features: Model your software release process, Define the stages of your software release process, Rapidly release new features, and Adapt to your needs.

Bottom Window (ap-south-1.console.aws.amazon.com/codesuite/codepipeline/pipeline/new?region=ap-south-1):

- Header:** AWS Services navigation bar with S3 and EC2 icons, search bar, and user info (Mumbai, Nishant Singh).
- Breadcrumbs:** Developer Tools > CodePipeline > Pipelines > Create new pipeline.
- Form (Step 1: Choose pipeline settings):**
 - Pipeline name:** 'firstwebapp_pipeline' (100 characters max).
 - Pipeline type:** V2 (selected).
 - Execution mode:** Queued (Pipeline type V2 required) (selected).
 - Service role:** Not specified.
- Footer:** Step 1 of 5, CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates, Privacy, Terms, Cookie preferences.

The screenshot shows the AWS CodePipeline console interface for creating a new pipeline. The pipeline is titled "firstwebapp3d156".

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-ap-south-1-firstwebapp_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.

Notes:

- The first pipeline execution will fail if variables have no default values.

Advanced settings:

Source:

Step 3: Add build stage

Step 4: Add deploy stage

Step 5: Review

Source provider: GitHub (Version 1)

Grant AWS CodePipeline access to your GitHub repository. GitHub to your pipeline.

Note: The GitHub (Version 1) action is not recommended. The selected action uses OAuth apps to authenticate. Instead, choose the GitHub (Version 2) action. GitHub Connections use GitHub Apps to manage authentication.

Change detection options:

Choose a detection mode to automatically start your pipeline.

GitHub webhooks (recommended): Use webhooks in GitHub to automatically start my pipeline when a change occurs.

Processing OAuth request:

Choose Confirm to complete the connection.

Buttons: Cancel, Confirm

Footer:

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS CodePipeline console interface for creating a new pipeline. The current step is 'Source'. A success message at the top indicates 'You have successfully configured the action with the provider.' Below it, a warning box states: 'The GitHub (Version 1) action is not recommended. The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more'.

Source provider: GitHub (Version 1)

Repository: HimanshuSingh1706/aws-elastic-beanstalk-express-js-sample

Branch: main

Change detection options:

- GitHub webhooks (recommended)
Use webhooks in GitHub to automatically start my pipeline when a change occurs
- AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Buttons: Cancel, Previous, Next

The screenshot shows the AWS CodePipeline 'Create new pipeline' wizard at Step 4: Add build stage. The left sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main panel is titled 'Build - optional' and contains the following fields:

- Build provider:** AWS CodeBuild
- Region:** Asia Pacific (Mumbai)
- Project name:** firstwebapp
- Environment variables - optional:** A search bar and a 'Create project' button.
- Build type:** Single build (selected) vs Batch build

At the bottom right of the main panel are 'Cancel', 'Previous', 'Skip build stage', and 'Next' buttons.

The screenshot shows the AWS CodePipeline 'Create new pipeline' wizard at Step 5: Add deploy stage. The left sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main panel is titled 'Deploy - optional' and contains the following fields:

- Deploy provider:** AWS Elastic Beanstalk
- Region:** Asia Pacific (Mumbai)
- Application name:** firstwebapp
- Environment name:** Firstwebapp-env

At the bottom right of the main panel are 'Cancel', 'Previous', 'Skip build stage', and 'Next' buttons.

The screenshot shows the AWS CodePipeline console with a success message: "Congratulations! The pipeline firstwebapp_pipeline has been created." The pipeline type is V2 and the execution mode is QUEUED. The Source stage is listed as succeeded. A button labeled "Disable transition" is visible at the bottom of the stage card.

Step 5 pipeline test

The screenshot shows the AWS CodePipeline console with the pipeline in progress. The Build stage (AWS CodeBuild) is listed as succeeded, and the Deploy stage (AWS Elastic Beanstalk) is in progress. A button labeled "Disable transition" is visible at the bottom of the stage card.