

Step 1: Create a deployment environment

The screenshots illustrate the step-by-step process of creating a new IAM role:

- Screenshot 1: IAM Roles List**
Shows the 'Roles' list with two existing roles: 'AWSServiceRoleForSupport' and 'AWSServiceRoleForTrustedAdvisor'. A 'Create role' button is visible.
- Screenshot 2: Step 1 - Select trusted entity**
Shows the 'Select trusted entity' step. Under 'Trusted entity type', 'AWS service' is selected. Under 'Use case', 'EC2' is selected. A 'Next Step' button is visible.
- Screenshot 3: Step 2 - Add permissions**
Shows the 'Add permissions' step. Under 'Permissions policies', the 'awselasticbeanstalk*' filter is applied, showing 14 matches. The 'AWSElasticBeanstalkWebTier' policy is selected.

Screenshot 1: AWS IAM - Roles

The screenshot shows the AWS IAM Roles page. A success message at the top says "Role exp-role created." The table lists three roles:

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
exp-role	AWS Service: ec2	-

Screenshot 2: AWS Elastic Beanstalk - Configure environment

The screenshot shows the "Configure environment" step of the Elastic Beanstalk wizard. The left sidebar lists steps 1 through 6. Step 1 is selected: "Configure environment". The main panel shows:

- Environment tier:** Web server environment (selected).
- Application information:** Application name: exp4elasticbeanstalk.
- Environment information:** Environment name: Exp4elasticbeanstalk-env.

Screenshot 3: AWS Elastic Beanstalk - Environment information

The screenshot shows the "Environment information" step of the Elastic Beanstalk wizard. The main panel shows:

- Environment name:** Exp4elasticbeanstalk-env.
- Domain:** Leave blank for autogenerated value: us-east-1.elasticbeanstalk.com. A "Check availability" button is next to it.
- Environment description:** An empty text area.
- Platform:** Platform type: Managed platform (selected). Platform: PHP.

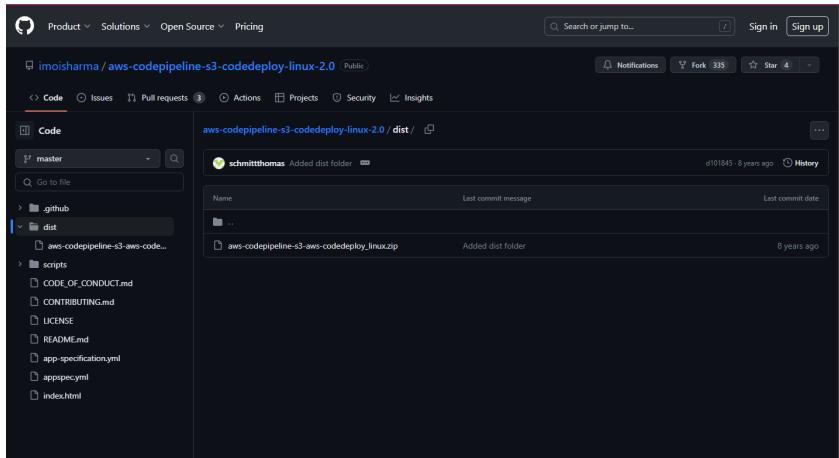
The screenshots illustrate the configuration steps for creating a new application environment in AWS Elastic Beanstalk:

- Step 1: Platform info**
 - Platform type:** Managed platform (selected)
 - Platform:** PHP
 - Platform branch:** PHP 8.2 running on 64bit Amazon Linux 2023
 - Platform version:** 4.0.0 (Recommended)
- Step 2: Configure service access**
 - 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.
 - Service role:** Create and use new service role (selected)
 - Service role name:** aws-elasticbeanstalk-service-role
 - EC2 key pair:** Choose an EC2 key pair to securely log in to your EC2 instances.
 - EC2 instance profile:** Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
- Step 3 - optional: Set up networking, database, and tags**
 - Virtual Private Cloud (VPC):** Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console.
 - Instance settings:** Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.
 - Public IP address:** Assign a public IP address to the Amazon EC2 instances in your environment. Activated.

The screenshots illustrate the configuration of an AWS Elastic Beanstalk environment across three steps:

- Step 1: Configure environment**
 - Throughput:** Set to 125 MiB/s.
 - Instance metadata service (IMDS):** Set to Deactivated.
 - EC2 security groups:** Default group selected.
 - Capacity Info:** Configure compute capacity and auto scaling settings.
- Step 2: Configure service access**
 - Architecture:** x86_64 selected.
 - Instance type:** t3.micro and t3.small selected.
 - AMI ID:** ami-05405593c57273c3 selected.
 - Availability Zones:** Any selected.
 - Placement:** Choose Availability Zones (AZs) dropdown.
- Step 3: Review**
 - Review Info:** Environment tier set to Web server environment, Application name to exp4elasticbeanstalk, Application code to Sample application.
 - Service access Info:** Service role set to am:aws:iam:928565427545:service-role/aws-elasticbeanstalk-.

Step 2: Get the copy of Sample Code



Open Amazon S3 console and Create your S3 Bucket:

The screenshots illustrate the process of creating an S3 bucket. The first part shows the overall S3 service page with a prominent 'Create a bucket' button. The second part shows the detailed 'Create bucket' wizard, where the user has entered the bucket name 'srushti' and selected the 'us-east-1' region. The 'General configuration' section is active, and the 'Object Ownership' section shows that ACLs are disabled.

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable
 Enable

Tags (0) - optional

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

adlexp4

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account.
Access to this bucket and its objects is specified using

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be

aws Services Search [Alt+S] Global Srushti Sanjay Sakharkar View details X ⓘ

Successfully created bucket "adlexp4"
To upload files and folders, or to configure additional bucket settings choose [View details](#).

Amazon S3 > Buckets

▶ Account snapshot
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

Buckets (1) Info

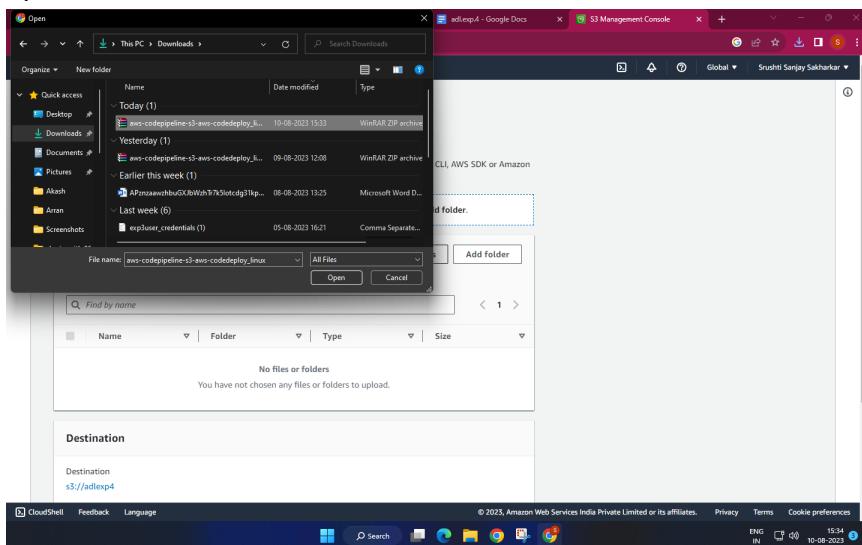
Buckets are containers for data stored in S3. [Learn more](#)

Copy content Empty Delete Create bucket

Find buckets by name

Name	AWS Region	Access	Creation date
adlexp4	US East (N. Virginia) us-east-1	Bucket and objects not public	August 10, 2023, 15:31:05 (UTC+05:30)

Upload the code to Bucket



Add Files -- upload zip files from downloads of your computer. Click on Upload Button

Summary	Status	
Destination s3://adlexp4	Succeeded 1 file, 5.8 KB (100.00%)	Failed 0 files, 0 B (0%)

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
aws-codedpipeline-s3-aws-codeddeploy_linux.zip	zip	August 10, 2023, 15:35:17 (UTC+05:30)	5.8 KB	Standard

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Step 3: Create Pipeline

Search results for 'elastic'

Try searching with longer queries for more relevant results

Services (12)

- Features (27)
- Resources (New) (1,777)
- Documentation (23,359)
- Knowledge Articles (20)
- Tutorials (18)
- Events (22)
- Marketplace (1,080)

Services

- Elastic Transcoder
- Elastic Beanstalk
- Elastic Container Service
- Elastic Container Registry

Features

- Elastic IPs
- EC2 feature

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

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. [Learn more](#)

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.

Benefits and features

Getting started

The screenshot shows the AWS CodePipeline Pipelines page. The left sidebar has a 'CodePipeline' section with links for Source, Artifacts, Build, Deploy, Pipeline, Pipelines (which is selected), and Settings. The main area shows a table with columns for Name, Most recent execution, Latest source revisions, and Last executed. A search bar at the top right is empty, and there are buttons for Create pipeline, Notify, View history, Release change, and Delete pipeline.

This is Step 1 of the pipeline creation wizard. It's titled 'Choose pipeline settings'. The pipeline name is set to 'exp4pipeline'. Under 'Service role', the 'New service role' option is selected, with a note: 'Create a service role in your account'. The 'Role name' field contains 'AWSCodePipelineServiceRole-us-east-1-exp4pipeline'. A checkbox for 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline' is checked. There is also an 'Advanced settings' link.

This is Step 2 of the pipeline creation wizard. It's titled 'Add source stage'. Under 'Source provider', 'Amazon S3' is selected. The 'Bucket' field contains 'adlexp4'. The 'S3 object key' field contains 'aws-codedeploy-s3-aws-codedeploy_linux.ipk'. Under 'Change detection options', the 'Amazon CloudWatch Events (recommended)' option is selected, with a note: 'Use Amazon CloudWatch Events to automatically start your pipeline when a change occurs in the source code.' There is also an 'AWS CodePipeline' option. Navigation buttons 'Cancel', 'Previous', and 'Next' are at the bottom.

Review the settings and create the pipeline.

Step 1: Choose pipeline settings

Pipeline name: exp4pipeline
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-exp4pipeline

Step 2: Add source stage

Source action provider: Amazon S3
PollForSourceChanges: false
S3Bucket: adflex4
S3ObjectKey: aws-codepipeline-s3-aws-codedeploy_linux.zip

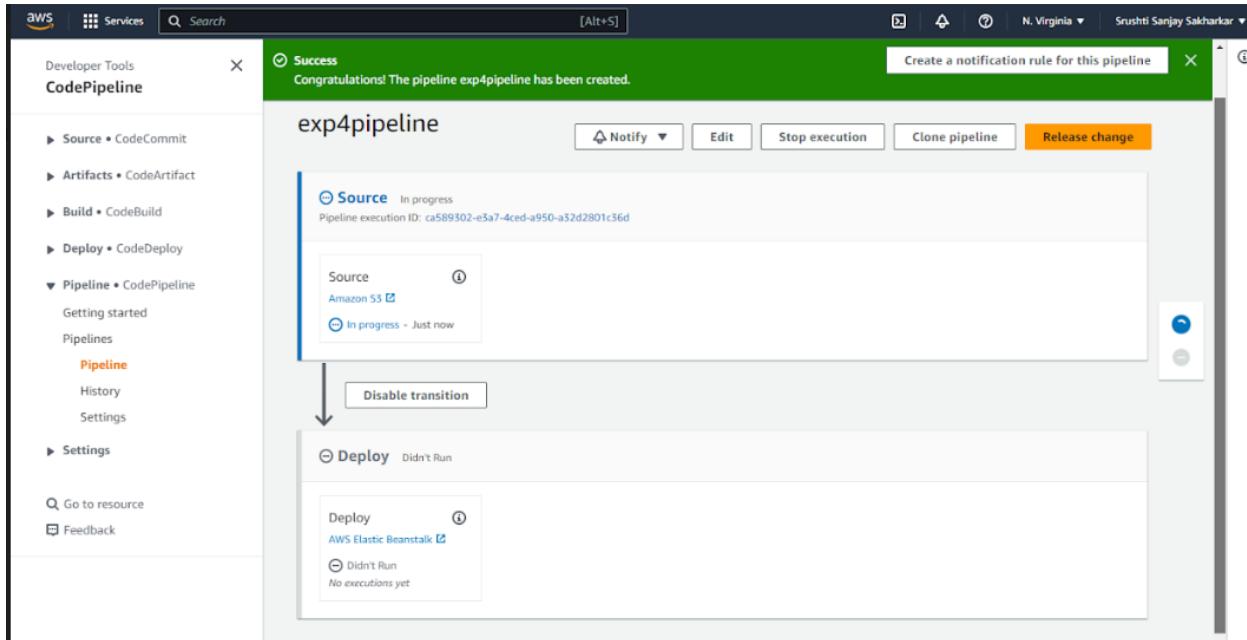
Step 3: Add build stage

Build action provider: No build

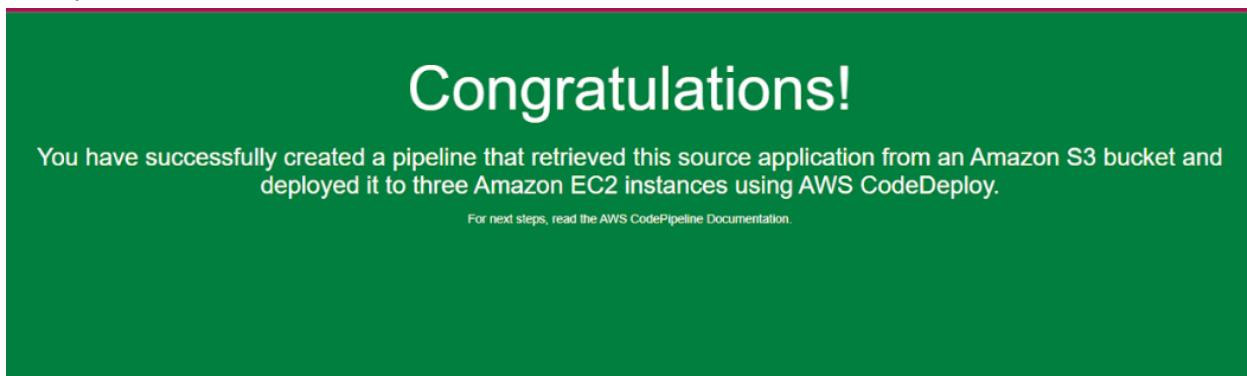
Step 4: Add deploy stage

Deploy action provider: AWS Elastic Beanstalk
ApplicationName: exp4elasticbeanstalk
EnvironmentName: Exp4elasticbeanstalk-env

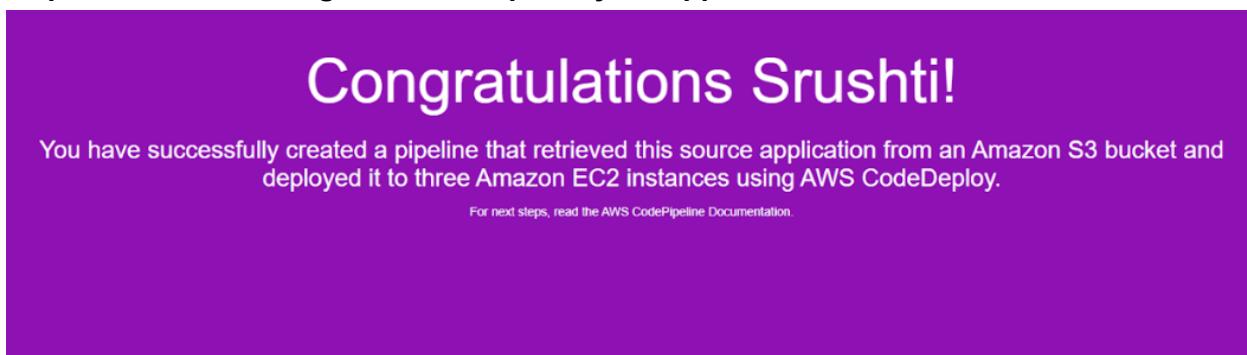
Cancel Previous Create pipeline



Now go to your EBS environment and click on the URL to view the sample website you deployed.



Step 5: Commit a change and then update your app



Step 6: Clean up the resources

⌚ Successfully emptied bucket "adlexp4"
View details below. If you want to delete this bucket, use the [delete bucket configuration](#).

Empty bucket: status

The details below are no longer available after you navigate away from this page.

Summary		
Source	Successfully deleted ⌚ 2 objects, 6.5 KB	Failed to delete 0 objects

Failed to delete (0)

Name	Prefix	Version ID	Type	Last modified	Size	Error
No failed object deletions						

⌚ Successfully deleted bucket "adlexp4"

Amazon S3

▶ Account snapshot
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

Buckets (2) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
codepipeline-us-east-1-451743391261	US East (N. Virginia) us-east-1	Bucket and objects not public	August 10, 2023, 16:04:08 (UTC+05:30)
elasticbeanstalk-us-east-1-928565427545	US East (N. Virginia) us-east-1	Objects can be public	August 10, 2023, 15:45:41 (UTC+05:30)