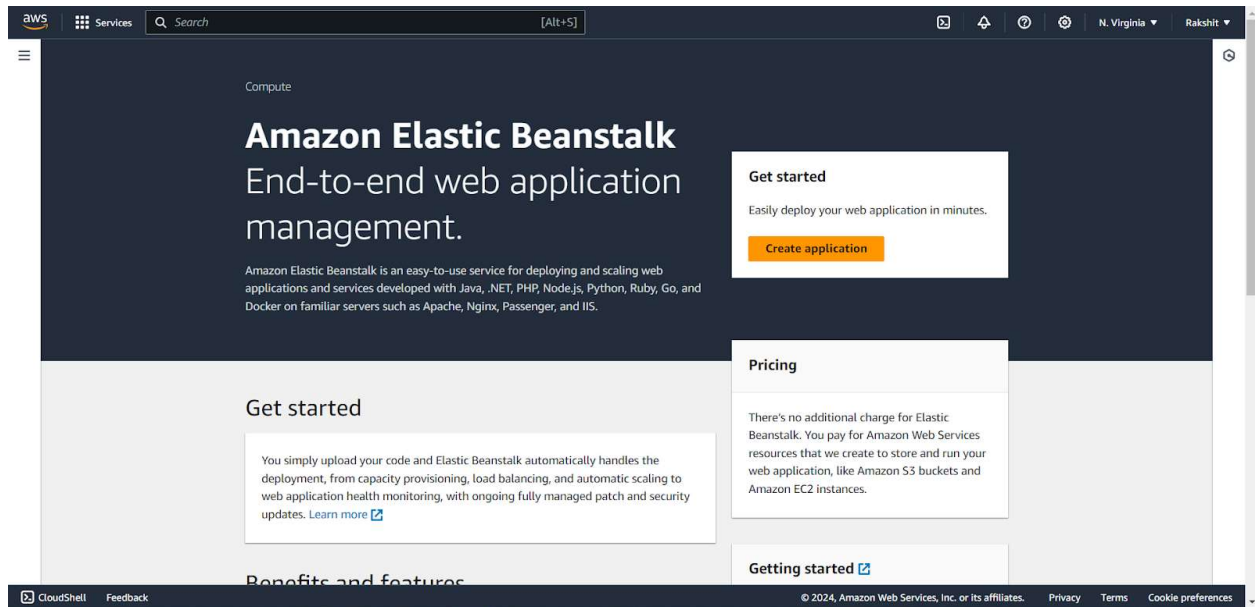
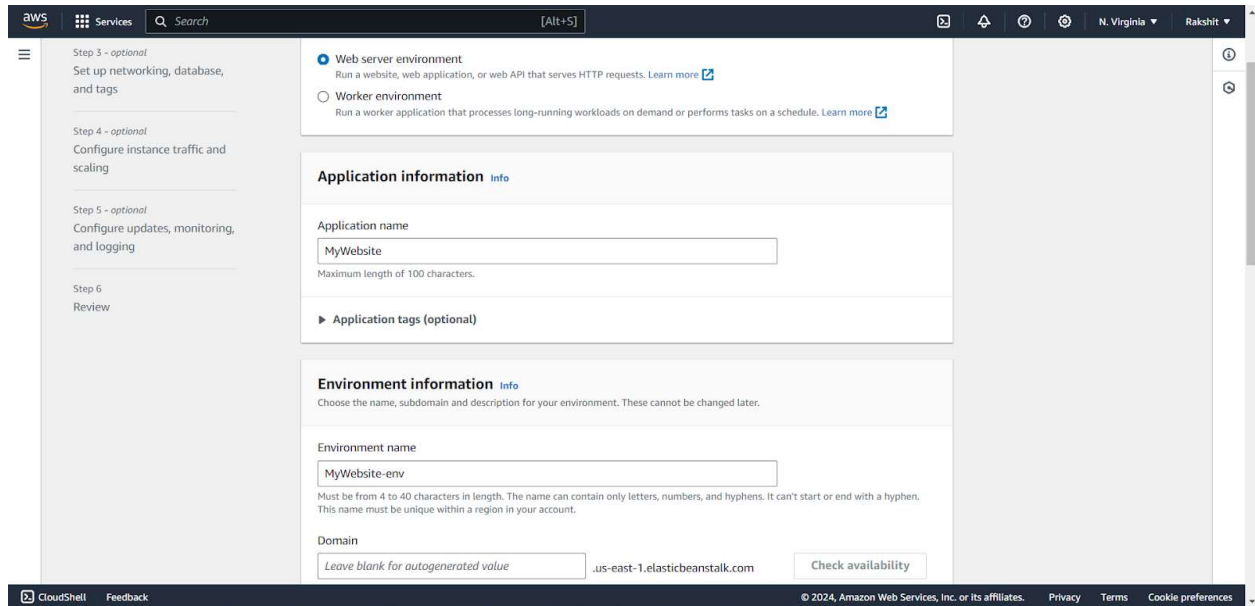


Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

1. Login to your AWS account and search for Elastic Beanstalk.



2. Click on create application. Enter your application names and other basic details.



3. In the platform select PHP among other options. Platform branch and platform version will be entered automatically.

The screenshot shows the 'Platform' configuration page in the AWS console. It has a title 'Platform' with an 'Info' link. Under 'Platform type', 'Managed platform' is selected with a radio button. Below it, a description states: 'Platforms published and maintained by Amazon Elastic Beanstalk. Learn more'. 'Custom platform' is unselected. Under 'Platform', a dropdown menu shows 'PHP'. Under 'Platform branch', a dropdown menu shows 'PHP 8.3 running on 64bit Amazon Linux 2023'. Under 'Platform version', a dropdown menu shows '4.3.2 (Recommended)'.

4. Keep the other setting to default and click on next. In service access click on “Use an existing service role”.

The screenshot shows the 'Configure service access' page in the AWS console. It has a title 'Configure service access' with an 'Info' link. On the left, a sidebar shows the setup steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled 'Service access' and contains the following sections: 'Service role' with radio buttons for 'Create and use new service role' and 'Use an existing service role' (the latter is selected); 'Existing service roles' with a dropdown menu showing 'aws-elasticbeanstalk-service-role' and a refresh button; 'EC2 key pair' with a dropdown menu showing 'Choose a key pair' and a refresh button; and 'EC2 instance profile' with a dropdown menu showing an empty field and a refresh button. At the bottom, there are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'.

5. Go to EC2 service and click on Key pair to create a new key pair. Give the key pair a name and select the type as RSA. For private key file format select .pem.

The screenshot shows the 'Create key pair' page in the AWS Management Console. The page title is 'Create key pair'. Below the title, there is a description: 'A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.' The form has three main sections: 'Name', 'Key pair type', and 'Private key file format'. The 'Name' field contains 'rakshit'. The 'Key pair type' section has two radio buttons: 'RSA' (selected) and 'ED25519'. The 'Private key file format' section has two radio buttons: '.pem' (selected) and '.ppk'. Below these sections, there is a 'Tags - optional' section with the text 'No tags associated with the resource.' and an 'Add new tag' button. At the bottom of the form, there are 'Cancel' and 'Create key pair' buttons.

4. Come back to Elastic Beanstalk configuration. Select the newly created key pair from the dropdown menu. Also select the EC2 instance profile. Click on next.

The screenshot shows the 'Configure service access' page in the AWS Management Console. The page title is 'Configure service access'. On the left, there is a sidebar with a list of steps: 'Step 1: Configure environment', 'Step 2: Configure service access' (current step), 'Step 3 - optional: Set up networking, database, and tags', 'Step 4 - optional: Configure instance traffic and scaling', 'Step 5 - optional: Configure updates, monitoring, and logging', and 'Step 6: Review'. The main content area has a 'Service access' section with a description: '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'. Below this, there are three sections: 'Service role' with radio buttons 'Create and use new service role' and 'Use an existing service role' (selected); 'Existing service roles' with a dropdown menu showing 'aws-elasticbeanstalk-service-role'; 'EC2 key pair' with a dropdown menu showing 'rakshit'; and 'EC2 instance profile' with a dropdown menu showing 'role1'. At the bottom, there are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

5. Skip to review. Review all the configurations and click on submit. Wait for the “Environment successfully launched” message.

The image shows two screenshots of the AWS Elastic Beanstalk console. The top screenshot displays the configuration page for a new environment. The bottom screenshot shows the environment successfully launched with a green status bar.

Platform software configuration:

Property	Value
Lifecycle	false
Log streaming	Deactivated
Allow URL fopen	On
Display errors	Off
Document root	-
Max execution time	60
Memory limit	256M
Zlib output compression	Off
Proxy server	nginx
Logs retention	7
Rotate logs	Deactivated
Update level	minor
X-Ray enabled	Deactivated

Environment properties:

No environment properties defined

Environment successfully launched message:

Environment successfully launched.

MyWebsite-env overview:

Property	Value
Health	Pending
Environment ID	e-xipkejthr8
Domain	MyWebsite-env.eba-upbp4pjs.us-east-1.elasticbeanstalk.com
Application name	Mywebsite

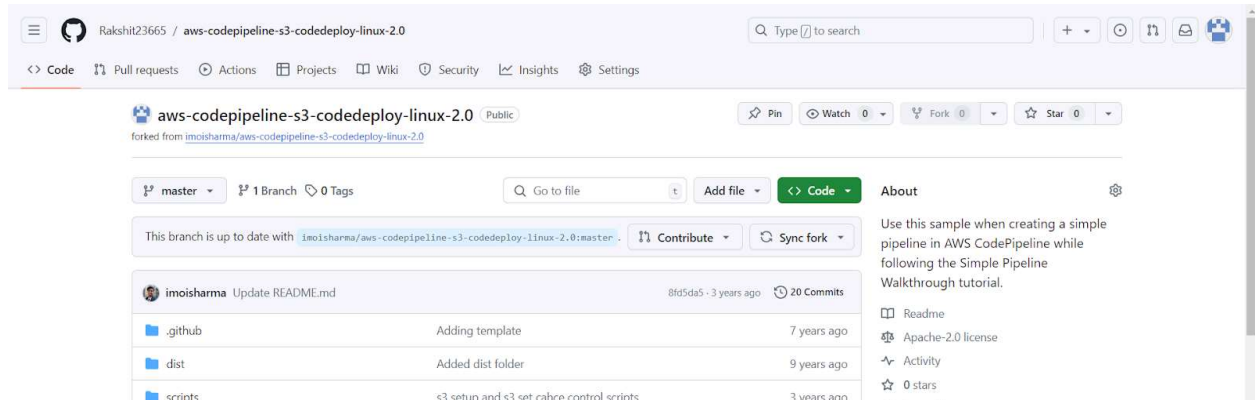
Platform:

Property	Value
Platform	PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2
Running version	-
Platform state	Supported

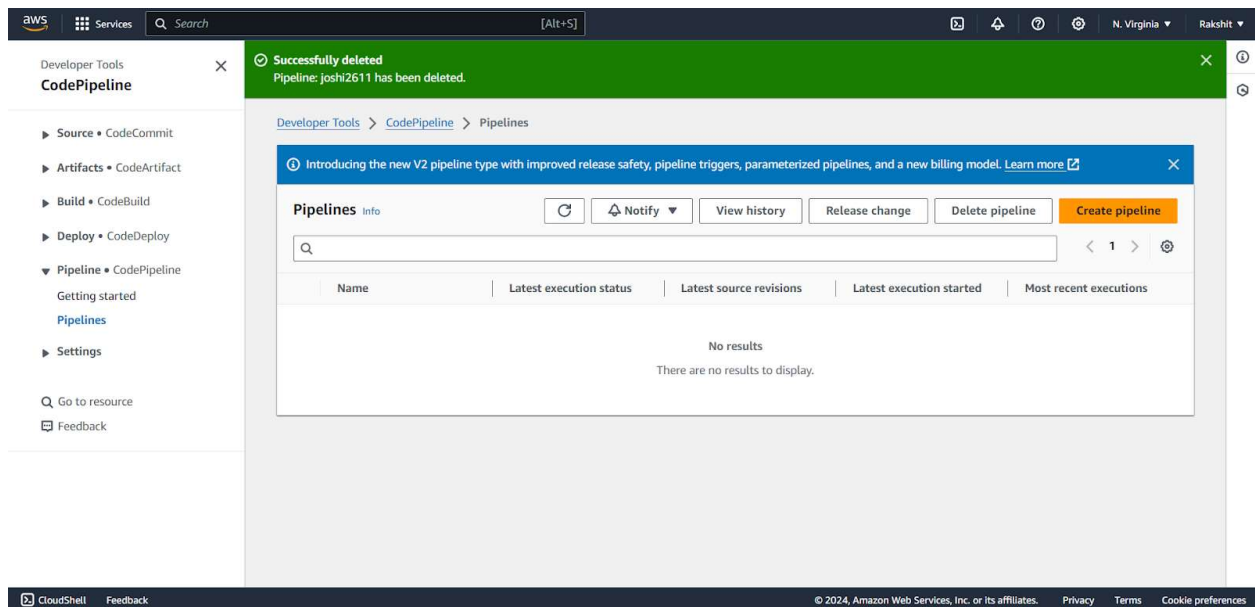
Events (10):

Filter events by text, property or value

6. Create a github repository with the source code to be deployed. Here I have forked an existing repository.



7. Go to CodePipeline service and click on create pipeline.



8. Give the pipeline a name. Role name will be generated automatically based on pipeline name

Choose pipeline settings [Info](#)

Step 1 of 5

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Pipeline type

ⓘ You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.

☐ Superseded
A more recent execution can overtake an older one. This is the default.

☒ Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

☐ Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

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

Type your service role name:

☒ Allow AWS CodePipeline to create a service role so it can be used with this new 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.

You can add up to 50 variables.

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

► **Advanced settings**

Cancel

Next

9. Select Github(Version 2) as a source provider. Click on connect to github to create a new connection if you don't have one.

The screenshot shows the AWS CodePipeline console interface for adding a source stage. The left sidebar contains a navigation menu with steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review), and a Review link. The main content area is titled 'Add source stage' with an 'Info' link and indicates 'Step 2 of 5'. The 'Source' section includes a 'Source provider' dropdown menu with 'GitHub (Version 2)' selected. Below this is an informational box for 'New GitHub version 2 (app-based) action' explaining that a connection is required. The 'Connection' section has a search input and a 'Connect to GitHub' button. The 'Repository name' section has a search input and a note about the format 'group/subgroup/project'. The 'Default branch' section has a search input. The 'Output artifact format' section has a dropdown menu with 'CodePipeline defaults' selected. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. (© 2024).

aws Services Search [Alt+S] N. Virginia Rakshit

Step 1 Choose pipeline settings
Step 2 **Add source stage**
Step 3 Add build stage
Step 4 Add deploy stage
Step 5 Review

Step 2 of 5

Add source stage Info

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.
[Search] or **Connect to GitHub**

Repository name
Choose a repository in your GitHub account.
[Search]
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.
[Search]

Output artifact format
Choose the output artifact format:
CodePipeline defaults

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

10. Give the connection a name and click on Install a new app. After this click on install. Once installation is complete click on connect to establish a connection.

The screenshot displays the AWS Developer Tools console interface for creating a new connection. The main window is titled "Connect to GitHub" and shows the "GitHub connection settings" section. The "Connection name" field is filled with "sample". Below this, the "GitHub Apps" section provides instructions and a search bar containing the ID "53777842", with an "Install a new app" button. A modal window titled "Install on your personal account RakshitSharma" is open on the right, showing options for repository access and permissions. The "All repositories" option is selected, and the "Read access to issues and metadata" and "Read and write access to administration, code, commit statuses, pull requests, and repository hooks" permissions are checked. The modal includes "Install" and "Cancel" buttons. At the bottom of the main window, there is a "Tags - optional" section and a "Connect" button.

11. Once the connection is established, you will get a success message. Select the repository containing the source code. Also select the branch(usually master). Be sure to select the no filter option in Trigger section. Click on next.

The screenshot shows the AWS CodePipeline console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a user profile 'Rakshit'. The main content area is titled 'Review' and displays a 'New GitHub version 2 (app-based) action' setup screen. A green box with a checkmark indicates 'Ready to connect' with the message 'Your GitHub connection is ready for use.' Below this, there are sections for 'Connection' (showing an existing connection ID), 'Repository name' (set to 'Rakshit23665/aws-codepipeline-s3-codedeploy-linux-2.0'), 'Default branch' (set to 'master'), and 'Output artifact format' (set to 'CodePipeline default'). The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc.

Trigger

Trigger type

Choose the trigger type that starts your pipeline.

- ☒ **No filter**
Starts your pipeline on any push and clones the HEAD.
- ☐ **Specify filter**
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.
- ☐ **Do not detect changes**
Don't automatically trigger the pipeline.

12. Skip the build stage and directly go to deploy stage. Select Elastic Beanstalk as Deploy provider. Select the Elastic Beanstalk application name that we created earlier. Click on next once done.

The screenshot shows the AWS CodePipeline console interface. On the left, a sidebar indicates the current stage is 'Step 5: Review'. The main panel is titled 'Deploy' and contains the following configuration fields:

- Deploy provider:** A dropdown menu with 'AWS Elastic Beanstalk' selected.
- Region:** A dropdown menu with 'US East (N. Virginia)' selected.
- Input artifacts:** A text input field with a dropdown arrow, currently empty.
- Application name:** A search input field with 'Mywebsite' entered.
- Environment name:** A search input field with 'MyWebsite-env' entered.
- Configure automatic rollback on stage failure:** An unchecked checkbox.

At the bottom of the configuration panel are three buttons: 'Cancel', 'Previous', and 'Next' (highlighted in orange). The footer of the console shows '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

13. Review the configurations made and click on create pipeline.

The screenshot shows the AWS CodePipeline console interface. The main panel displays two steps in a list:

- Step 3: Add build stage**
 - Build action provider:** (Empty field)
 - Build stage:** A dropdown menu with 'No build' selected.
- Step 4: Add deploy stage**
 - Deploy action provider:** (Empty field)
 - Deploy action provider:** A dropdown menu with 'AWS Elastic Beanstalk' selected.
 - ApplicationName:** A text input field with 'Mywebsite' entered.
 - EnvironmentName:** A text input field with 'MyWebsite-env' entered.
 - Configure automatic rollback on stage failure:** A text input field with 'Disabled' entered.

At the bottom of the configuration panel are three buttons: 'Cancel', 'Previous', and 'Create pipeline' (highlighted in orange). The footer of the console shows '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

14. Once the pipeline is created you can go to the environments page(Elastic Beanstalk). The website is hosted on the link under domain column. Click on the link to go to the hosted website.

The screenshot shows the AWS Elastic Beanstalk console. On the left, the 'Environments' tab is selected under 'Applications'. The main area displays a table of environments. The first environment, 'MyWebsite-env', is highlighted. Below the table, the 'MyWebsite-env' details are shown, including the 'Domain' column which contains the URL 'mywebsite-env.eba-upbp4pjs-east-1.elasticbeanstalk.com'. Below the console, a browser window shows the hosted website. The website has a green background with the text 'Rakshit Sharma d15c' in large white letters. Below this, it says 'You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.' and 'For next steps, read the AWS CodePipeline Documentation. Incodege 2020'.

Environment name	Health	Application	Platform	Domain	Running version	Tier name	Date created
MyWebsite-env	No Data	Mywebsite	PHP 8.3 runni...	MyWebsite-env.eba-upbp4pjs...	code-pipeline...	WebServer	August

Rakshit Sharma d15c

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation. Incodege 2020