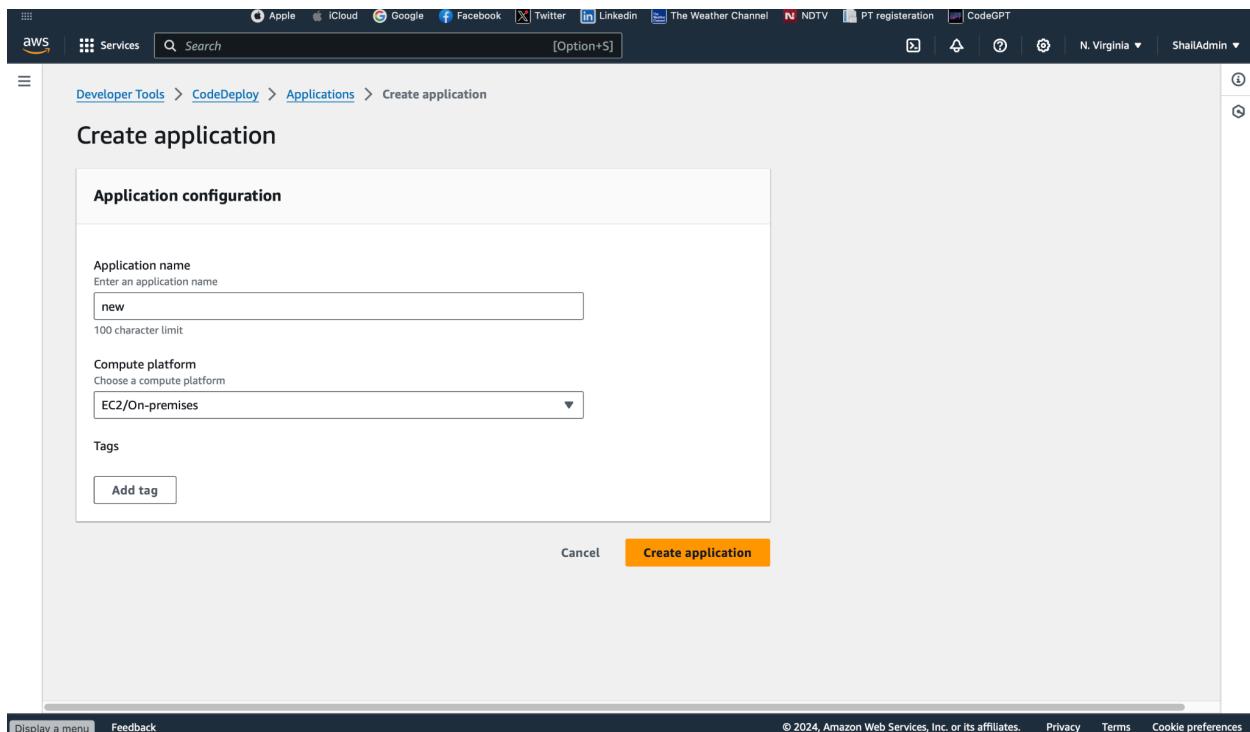


The screenshot shows the AWS CodePipeline console. On the left, a sidebar menu lists various services under 'Developer Tools'. The 'CodePipeline' section is expanded, showing 'Source', 'Artifacts', 'Build', 'Deploy', and 'Pipeline'. Under 'Pipeline', 'Getting started', 'Pipelines' (which is selected), and 'Settings' are listed. Below the sidebar are links for 'Go to resource' and 'Feedback'. The main content area displays a 'Pipelines' table with one item: 'deml-pipeline'. The table columns include Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. The pipeline name is 'deml-pipeline', its status is 'Failed', it was triggered by 'Source - fdb20af1: demo data', and it started 1 month ago. A 'View details' link is also present. At the top of the main area, a banner introduces the new V2 pipeline type. Navigation icons like back, forward, and search are at the top right.

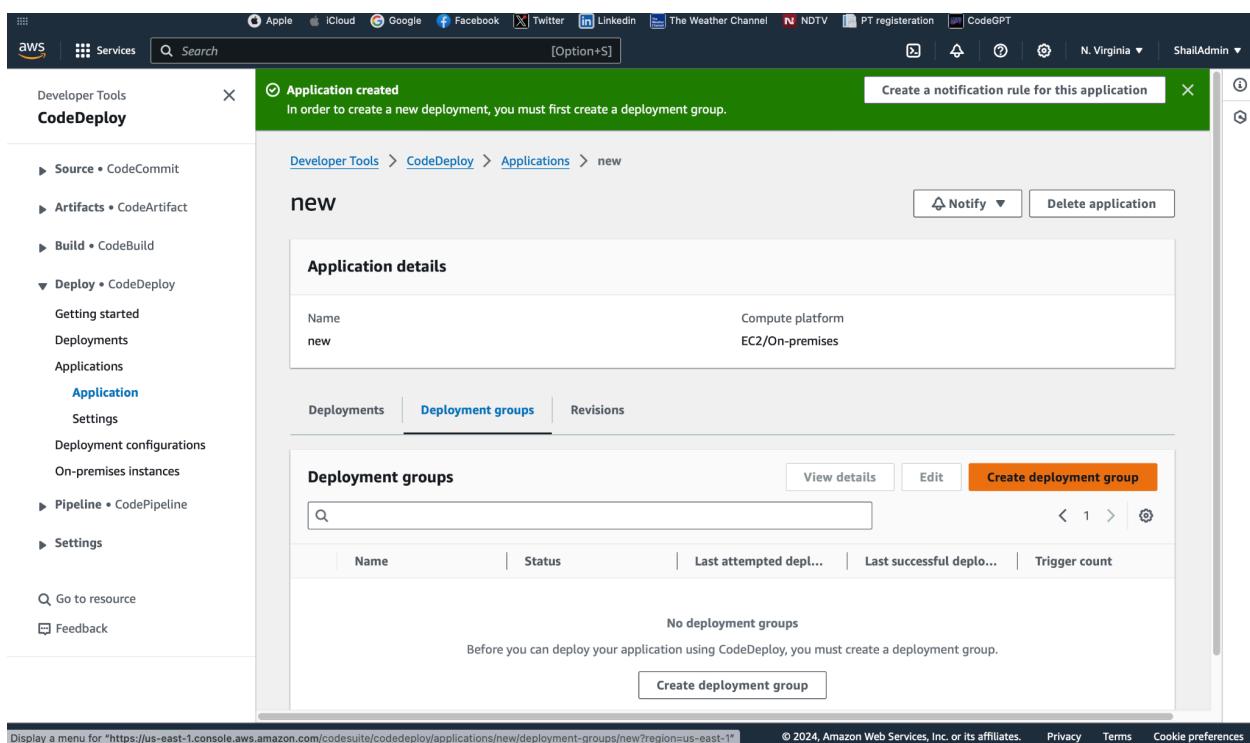
AWS CodeCommit is deprecated. Go to Code Deploy

The screenshot shows the AWS CodeDeploy console. The sidebar menu is identical to the previous screenshot, with 'CodeDeploy' selected under 'Pipeline'. The main content area has a dark background. In the center, there's a large heading 'AWS CodeDeploy' followed by the subtext 'Automate code deployments to maintain application uptime'. To the right, a white callout box contains the text 'Create AWS CodeDeploy deployment' and 'Get started with AWS CodeDeploy by creating your first deployment application.' It features a prominent orange 'Create application' button. Below this, a section titled 'How it works' includes a small image of a server icon with the text 'Introduction to AWS CodeDeploy - Aut...'. On the far right, a 'Pricing (US)' table provides cost information. The table has two rows: one for 'For CodeDeploy on EC2/Lambda' which is 'Free', and another for 'For CodeDeploy on Lambda' which costs '\$0.02 per instance'. The bottom of the page includes standard AWS footer links for 'Display a menu', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Click on Create Application



Choose application name and compute platform



Create Deployment Group

The screenshot shows the 'Create deployment group' wizard in the AWS CodeDeploy console. The application is named 'new' and the compute type is 'EC2/On-premises'. The deployment group name is 'test-deploy'. The service role ARN is 'arn:aws:iam::637423339839:role/deploy'.

Application
new
Compute type
EC2/On-premises

Deployment group name
Enter a deployment group name
test-deploy
100 character limit

Service role
Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.
arn:aws:iam::637423339839:role/deploy

Create service role

Ex-

The screenshot shows the AWS IAM Roles page. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access Analyzer, External access, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies). A "Related consoles" section at the bottom links to CloudWatch Metrics and CloudWatch Metrics Insights.

The main content area shows the "AWSCodePipelineServiceRole-us-east-1-test" role under the "AWSCodePipelineServiceRole-us-east-1-test" path. The "Summary" tab is selected, displaying details like Creation date (September 21, 2024, 23:10 UTC+05:30), Last activity (none), ARN (arn:aws:iam::637423339839:role/service-role/AWSCodePipelineServiceRole-us-east-1-test), and Maximum session duration (1 hour). The "Edit" button is located in the top right corner of the summary card.

The "Permissions" tab is active, showing one policy attached: "AWSCodePipelineServiceRole-us-ea...". It has a "Simulate" button, a "Remove" button, and a "Add permissions" dropdown. Below this, a table lists the attached policy with columns for Policy name, Type, and Attached entities.

The "Permissions boundary (not set)" section is shown below the policy table.

Choose deployment type

The screenshot shows the "Deployment type" configuration page. The left sidebar is identical to the one in the IAM screenshot, showing the same navigation options.

The main content area is titled "Deployment type" and "Choose how to deploy your application". It offers two options: "In-place" (selected) and "Blue/green". The "In-place" option describes updating instances in the deployment group with the latest application revisions. The "Blue/green" option describes replacing instances with new ones and deploying the latest revision to them.

The "Environment configuration" section allows selecting Amazon EC2 Auto Scaling groups or instances. The "Amazon EC2 instances" checkbox is checked, showing 0 unique matched instances. A link "Click here for details" is provided.

Below this, instructions for adding tags to EC2 instances are given, mentioning "One tag group": Any instance identified by the tag group will be deployed to, and "Multiple tag groups": Only instances identified by all the tag groups will be deployed to.

A "Tag group 1" section shows a key "Name" with a value "test" and a "Remove tag" button. An "Add tag" button is also present.

The bottom of the page includes standard AWS footer links: © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS Lambda console with the 'Deployment settings' tab selected. At the top, there are two radio button options: 'Only once' and 'Now and schedule updates'. The 'Now and schedule updates' option is selected. Below this, there are two tabs: 'Basic scheduler' (selected) and 'Cron expression'. A dropdown menu shows '14' and 'Days' is selected. Under 'Deployment settings', there is a section titled 'Deployment configuration' with a dropdown menu set to 'CodeDeployDefault.AllAtOnce' and a 'Create deployment configuration' button. In the 'Load balancer' section, there is a note about managing incoming traffic and a checkbox for 'Enable load balancing' which is unchecked. At the bottom right are 'Cancel' and 'Create deployment group' buttons.

Create deployment group. Disable load balancing.

The screenshot shows the AWS CodePipeline console with the 'Pipelines' page. On the left, a sidebar navigation includes 'Developer Tools', 'CodePipeline' (selected), 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline' (selected), 'Getting started', 'Pipelines' (selected), and 'Settings'. Below these are links for 'Go to resource' and 'Feedback'. The main content area shows a message about the new V2 pipeline type. It has a search bar and a table with columns: Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. One pipeline named 'deml-pipeline' is listed with a status of 'Failed' and a 'QUEUED' source. At the bottom right of the table is a 'View details' link. The top navigation bar includes the AWS logo, services like Apple, iCloud, Google, Facebook, Twitter, LinkedIn, The Weather Channel, NDTV, PT registration, and CodeGPT, a search bar, and user information for 'N. Virginia' and 'ShallAdmin'.

Create Pipeline

The screenshot shows the 'Choose pipeline settings' step of the AWS CodePipeline creation wizard. The pipeline name is set to 'test'. The pipeline type is 'V2', indicated by a note: '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.' The execution mode is 'Queued (Pipeline type V2 required)', which processes executions one by one. A service role is selected as 'New service role' with the name 'AWSCodePipelineServiceRole-us-east-1-test'. A checkbox at the bottom allows AWS CodePipeline to create a service role.

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

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

The screenshot shows the 'Add source stage' step of the AWS CodePipeline creation wizard. The source provider is set to 'GitHub (Version 1)'. A note indicates that this action is not recommended as it uses OAuth apps to access GitHub. Instead, it suggests using the GitHub (Version 2) action. Change detection options are set to 'GitHub webhooks (recommended)', which uses webhooks in GitHub to automatically start the pipeline when changes occur.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Add source stage Info

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Source

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

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

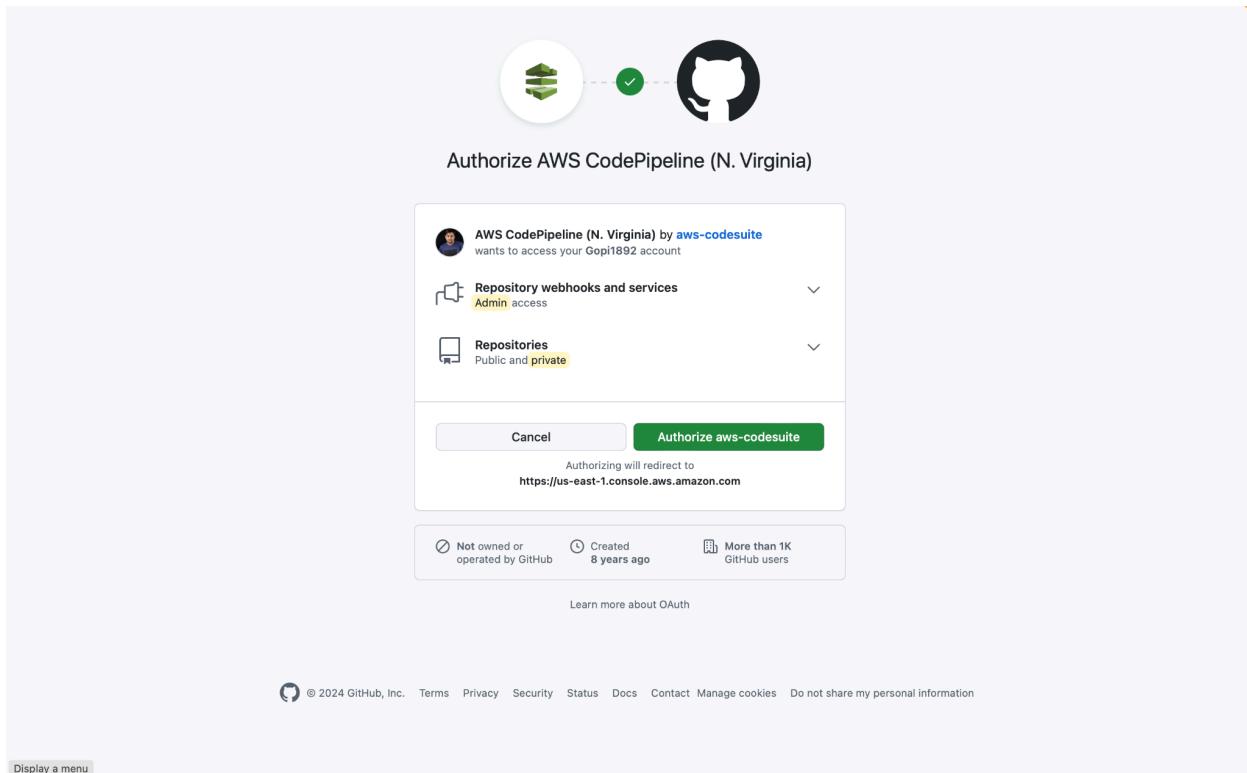
The GitHub (Version 1) action is not recommended
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

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

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



Connect to Github.

The screenshot shows the AWS CodePipeline configuration interface. On the left, a sidebar lists steps: Step 4 (Add deploy stage), Step 5 (Review). The main area shows a step titled "GitHub (Version 1)". A message box says "Connected" and "You have successfully configured the action with the provider." A warning message in a blue 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". Below this, there are fields for "Repository" (Gopi1892/Calculator-javaapp) and "Branch" (master). Under "Change detection options", there are two radio button options: "GitHub webhooks (recommended)" (selected) and "AWS CodePipeline". At the bottom, there are links for "Display a menu", "Feedback", and copyright information: "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Select Repo and branch.

Screenshot of the AWS CodePipeline 'Create new pipeline' wizard, Step 3: Add build stage.

The screenshot shows the 'Add build stage' configuration screen. 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), and Step 5 (Review). The main area is titled 'Build - optional'.

Build provider: AWS CodeBuild

Region: US East (N. Virginia)

Input artifacts: SourceArtifact (dropdown menu, currently empty)

Project name: demo-project (search bar) or Create project (button)

Environment variables - optional: Add environment variable (button)

Build type: Single build (radio button selected) / Batch build (radio button)

At the bottom, there are buttons: Cancel, Previous, Skip build stage, and Next (highlighted in orange).

AWS Services Search [Option+S] N. Virginia ShailAdmin

Choose pipeline settings Step 4 of 5

Add source stage

Add build stage

Add deploy stage

Step 5 Review

Deploy - optional

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. [Learn more](#)

BuildArtifact

No more than 100 characters

Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

new

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

test-deploy

Configure automatic rollback on stage failure

Cancel Previous Skip deploy stage Next

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

Select Application name and deployment group.

AWS Services Search [Option+S] N. Virginia ShailAdmin

Developer Tools **CodePipeline**

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
- Getting started
- Pipelines
- Pipeline**
- History
- Settings
- Settings
- Go to resource
- Feedback

Source
GitHub (Version 1) [\[More\]](#)

Succeeded
Sep 21, 2024 11:10 PM (UTC+5:30)
2fb32782 [\[More\]](#)

[View details](#)

[2fb32782 \[More\]](#) Source: Set up CI with Azure Pipelines [\[More\]](#)

[Disable transition](#)

Build Succeeded
Pipeline execution ID: [aeccc57-c428-48ff-a168-ffe47682f58c](#) [Start rollback](#)

Build
AWS CodeBuild

Succeeded
Sep 21, 2024 11:11 PM (UTC+5:30)
2fb32782 [\[More\]](#)

[View details](#)

[2fb32782 \[More\]](#) Source: Set up CI with Azure Pipelines [\[More\]](#)

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

The screenshot shows the AWS CodePipeline console. On the left, a sidebar menu is open under the 'CodePipeline' section, showing options like Source, Artifacts, Build, Deploy, Pipeline, History, Settings, and Settings. The main area displays a pipeline execution ID: [2fb32782](https://us-east-1.console.aws.amazon.com/codestar-codepipeline/home?#/executions/2fb32782). The execution details show a successful 'Build' step (AWS CodeBuild) and a failed 'Deploy' step (AWS CodeDeploy). The Deploy step failed on Sep 21, 2024 at 11:11 PM (UTC+5:30). There are buttons for 'Start rollback', 'Retry stage', and 'Retry failed actions'. A red vertical bar highlights the Deploy stage. On the far right, there are three circular status indicators: green, green, and red.

Deploy stage will fail as we don't have EC2 created.