

> sessions/workshops/



Continuous Delivery With GitHub Actions



Yadhav Jayaraman

Solutions Architect, GitHub
[@decyjphr](https://twitter.com/decyjphr)



@ppremk

Implementation Engineer



@iamhughes

DevOps Engineer

Continuous Delivery with GitHub Actions

Continuous Delivery

Continuous Delivery with GitHub Actions

“

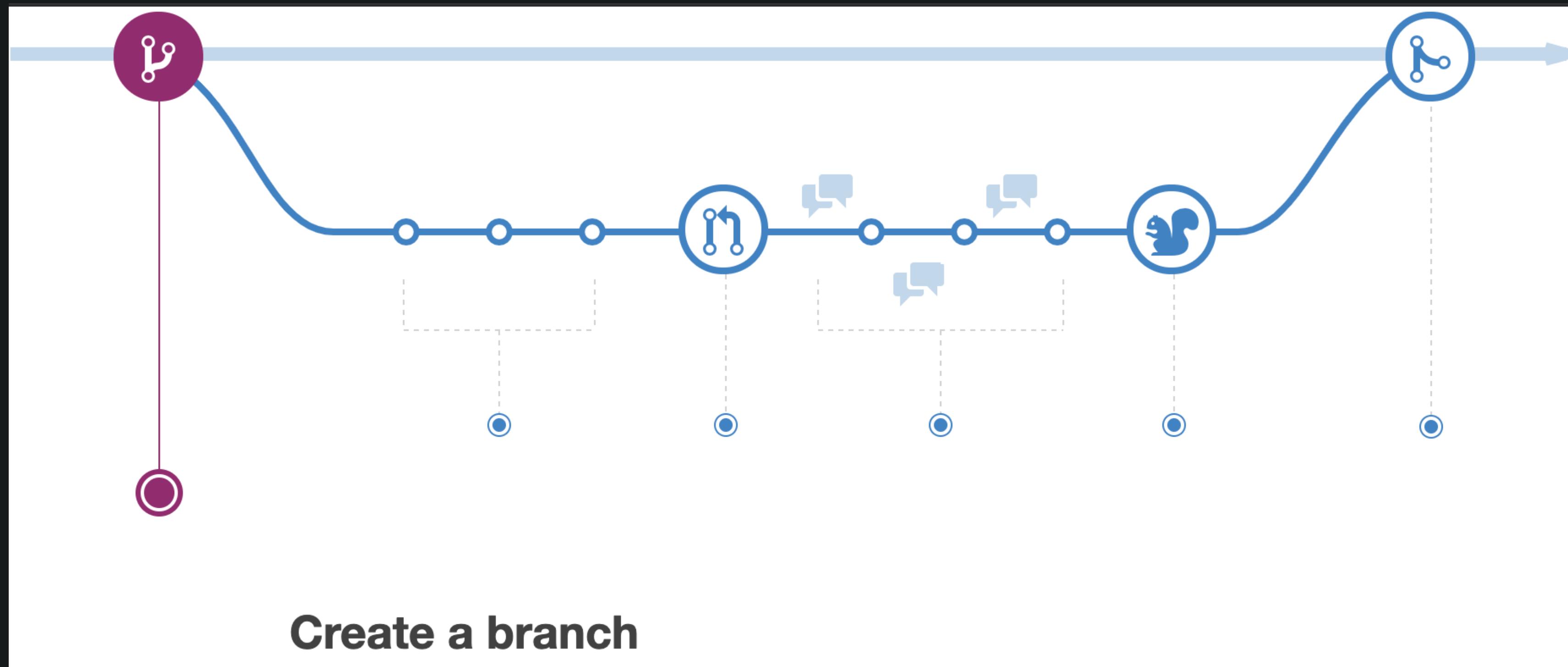
Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time.

– Martin Fowler

Continuous Delivery with GitHub Actions

Continuous Delivery with GitHub Actions

Understanding GitHub Flow



Hands-on Time

Pre-requisites

Github Account

Learning Lab Actions

GitHub Package Registry Personal Access Token

AWS Account

Permissions

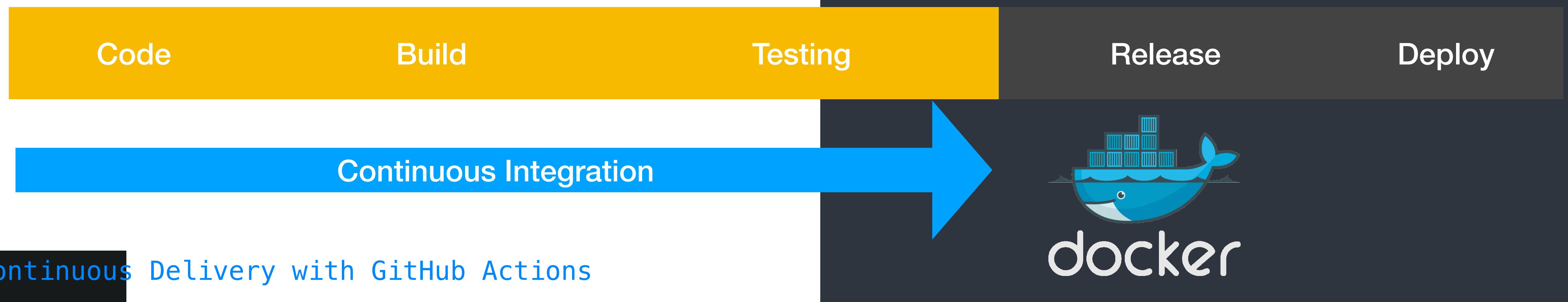
S3 Bucket

AWS Secret Key and Access Key

💡 Prerequisites

1. GitHub account with a [verified email address](#)
2. [Install GitHub Learning Lab](#) on your account
3. Sign up for the [GitHub Actions](#)
4. Sign up for the [GitHub Package Registry](#)
5. Create a [GitHub Personal Access Token](#) for your account.
6. [AWS Account](#)
 - The following permissions will be necessary for the second lab
 - Note: These are all `FullAccess`, which is not needed for production, but the permissions can be made more granular
 - `AmazonEC2FullAccess`
 - `CloudFormationAllAccess`
 - `AmazonAPIGatewayAdministrator`
 - `AmazonS3FullAccess`
 - `AWSLambdaFullAccess`
 - `IAMFullAccess`
7. Create an [AWS S3 bucket](#)
 - Note: You will need to sign up for S3 in addition to your normal account, if you haven't already
8. Both [AWS Secret key](#) and [AWS Access Key](#)

Using GitHub Actions to create and upload packages to the GitHub Package Registry after successful CI testing





Workshop 1

Complete the following steps:

- ✓ Create a CI Workflow
- ✓ Modify it to a CD Workflow
- ✓ Build the Docker Job
- ✓ Trigger the Workflow
- ✓ Verify the package



Course Steps

Course steps

- 1 **Move the workflow file**
Place the CI workflow file in the proper directory. Start
- 2 **Edit the workflow file**
Build the Docker workflow and 'secrets' configuration.
- 3 **Trigger the pipeline**
Make a commit and watch the pipeline work through GitHub Actions.
- 4 **View the workflow**
Verify whether the Docker workflow was successful.

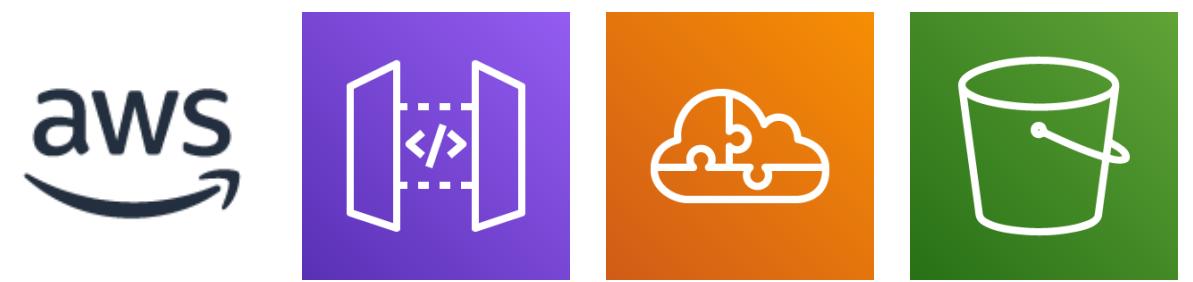
Course steps		...
✓	Move the workflow file	Completed 44 minutes ago
✓	Edit the workflow file	Completed 32 minutes ago
✓	Trigger the pipeline	Completed 30 minutes ago
✓	View the workflow	Completed 27 minutes ago

Register for *Publish to GitHub Packages*

1. Go to <https://git.io/Jfneg>
or go to [https://
lab.github.com/](https://lab.github.com/) and
search for “GitHub
Actions”
2. On the course landing
page, click **Join this
course**
3. Click **Start**

The screenshot shows a course landing page from the GitHub Learning Lab. At the top, there are navigation links: 'Learning Lab', 'For Organizations', and 'Teach on Learning Lab'. Below the header, the course title 'GitHub Actions: Publish to GitHub Packages' is displayed in large, bold, dark text. Underneath the title, it says 'The GitHub Training Team'. A brief description follows: 'This course will walk you through using GitHub Actions to get your code in a deployable state once your CI workflows have completed.' At the bottom of the page, there is a green button labeled 'Start free course' and the text 'Join 414 others!'

Using GitHub Actions to deploy an application to Amazon Web Services upon successful CI testing



Code

Build

Testing

Release

Deploy

Continuous Delivery



Workshop 2

Complete the following steps:

- ✓ Create a CD Workflow
- ✓ Job based on Label value
- ✓ Job to deploy to AWS
- ✓ Trigger the Workflow
- ✓ Verify the app is deployed



Course Steps

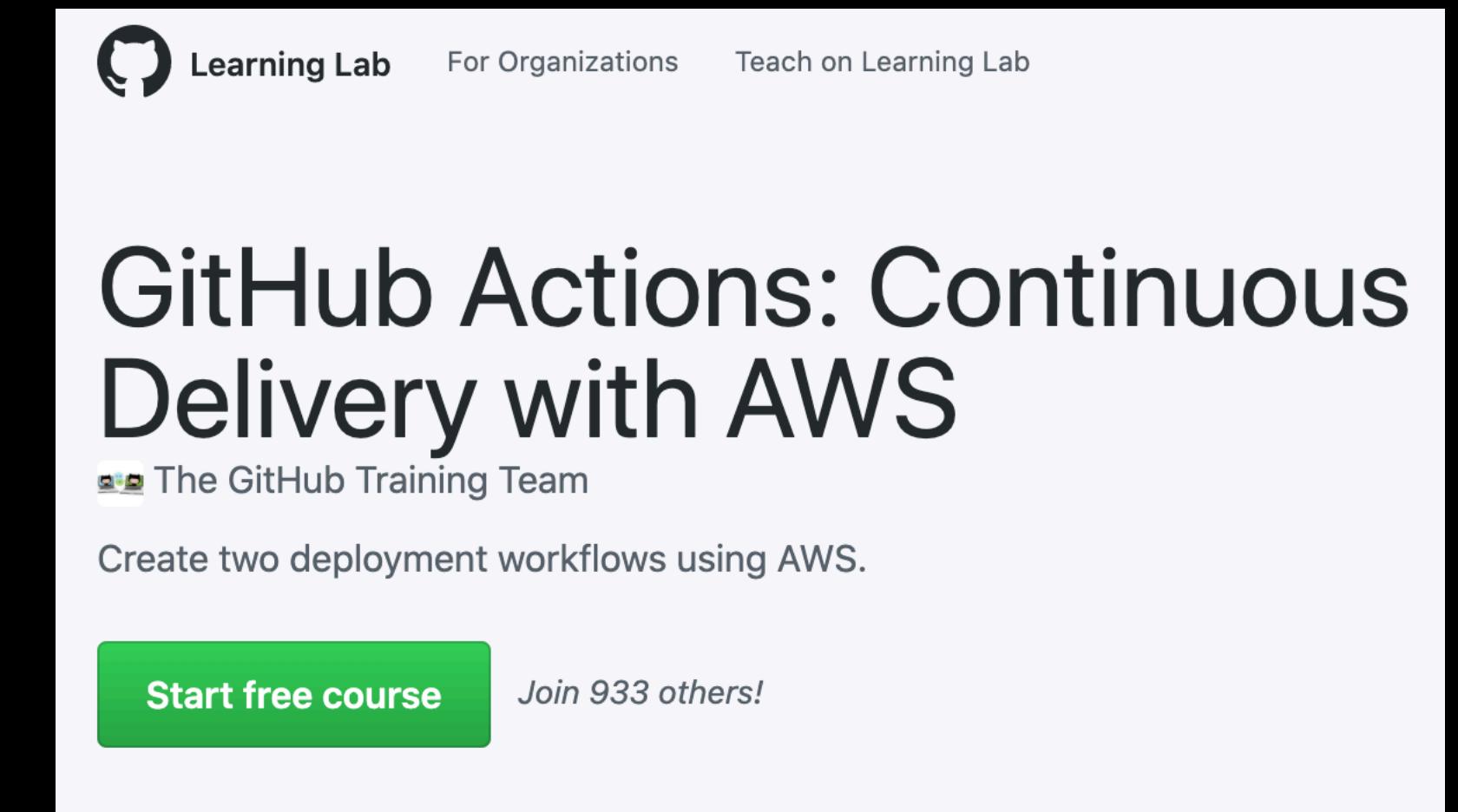
Course steps

- 1 **Configure a trigger based on labels**
Configure the workflow trigger based on an a label being added Start
- 2 **Trigger a job on specific labels**
Use a conditional to trigger a job on a specific label
- 3 **Deploy a Node.js app to AWS for the first time**
Use the GitHub-created action to deploy to AWS
- 4 **Merge the staging workflow**
Merge this staging workflow pull request
- 5 **Create an S3 bucket**
Create an S3 bucket in AWS and tell the action where to find it
- 6 **Approve pull request**
Approve pull request adding the aws-config.yml and sam-template.yml
- 7 **Test the staging action**
Test the new action to deploy labeled pull requests to staging
- 8 **Write the production deployment trigger**
Write the production deployment trigger in the new workflow
- 9 **Complete the deployment to production workflow**
Commit the steps to the production workflow that allow you to deploy on merge to master.
- 10 **Create the Docker image and push it to GitHub Packages**
Add a step to the workflow for building and pushing a Docker image to GitHub Packages

Course steps		...
1	Configure a trigger based on labels Configure the workflow trigger based on an a label being added	<button>Start</button>
2	Trigger a job on specific labels Use a conditional to trigger a job on a specific label	
3	Deploy a Node.js app to AWS for the first time Use the GitHub-created action to deploy to AWS	
4	Merge the staging workflow Merge this staging workflow pull request	
5	Create an S3 bucket Create an S3 bucket in AWS and tell the action where to find it	
6	Approve pull request Approve pull request adding the aws-config.yml and sam-template.yml	
7	Test the staging action Test the new action to deploy labeled pull requests to staging	
8	Write the production deployment trigger Write the production deployment trigger in the new workflow	
9	Complete the deployment to production workflow Commit the steps to the production workflow that allow you to deploy on merge to master.	
10	Create the Docker image and push it to GitHub Packages Add a step to the workflow for building and pushing a Docker image to GitHub Packages.	
11	Merge the production workflow Merge this pull request and test the production deployment workflow	

Register for *Continuous Delivery with AWS*

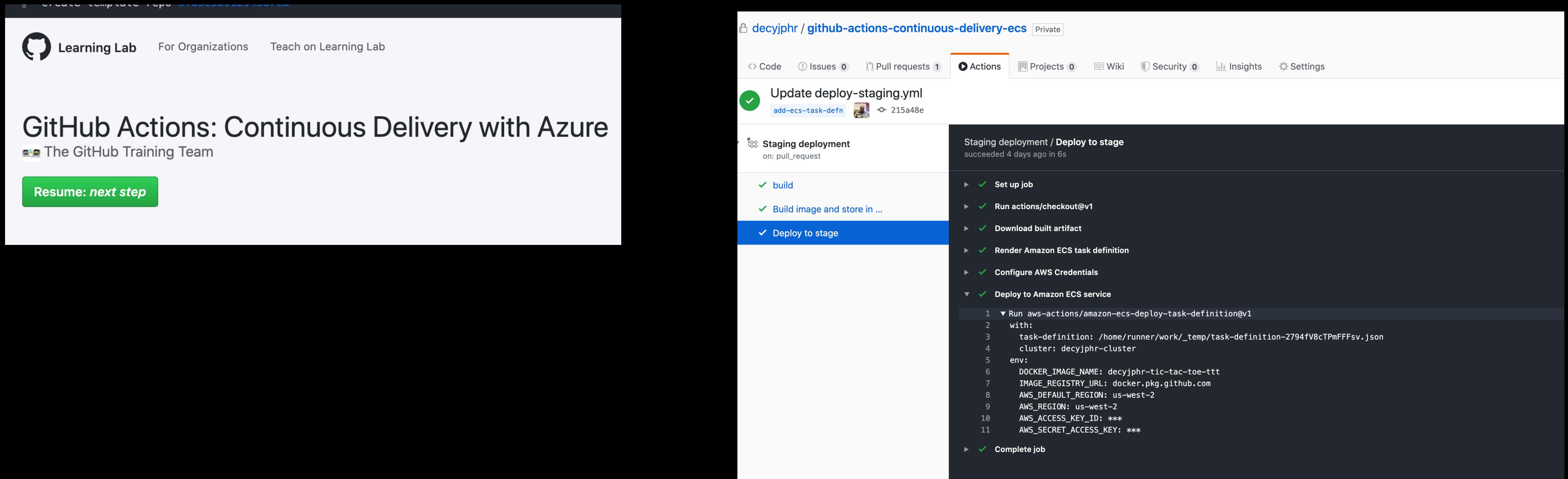
1. Go to <https://git.io/JfZNF>
or go to [https://
lab.github.com/](https://lab.github.com/) and
search for “GitHub
Actions”
2. Choose the course
3. On the course landing
page, click **Join this
course**
4. Click **Start**



Extra

Continuous Delivery with GitHub Actions

Extra: Deploying Containers



The image displays two screenshots illustrating the deployment of containers using GitHub Actions.

Left Screenshot: A screenshot of the GitHub Learning Lab interface titled "GitHub Actions: Continuous Delivery with Azure". It shows a green button labeled "Resume: next step".

Right Screenshot: A screenshot of a GitHub repository named "decyjphr / github-actions-continuous-delivery-ecs". The "Actions" tab is selected, showing a successful run of "Update deploy-staging.yml". The "Staging deployment" job is listed, showing steps: "build", "Build image and store in ...", and "Deploy to stage". The "Deploy to stage" step is highlighted with a blue bar. To the right, the logs for the "Deploy to stage" step are displayed, detailing the deployment process to an Amazon ECS service. The logs show commands like "aws-actions/amazon-ecs-deploy-task-definition@v1" and environment variables such as DOCKER_IMAGE_NAME, IMAGE_REGISTRY_URL, AWS_DEFAULT_REGION, AWS_REGION, AWS_ACCESS_KEY_ID, and AWS_SECRET_ACCESS_KEY.

```
1   ▾ Run aws-actions/amazon-ecs-deploy-task-definition@v1
2     with:
3       task-definition: /home/runner/work/_temp/task-definition-2794fV8cTPmFFFsv.json
4       cluster: decyjphr-cluster
5       env:
6         DOCKER_IMAGE_NAME: decyjphr-tic-tac-toe-ttt
7         IMAGE_REGISTRY_URL: docker.pkg.github.com
8         AWS_DEFAULT_REGION: us-west-2
9         AWS_REGION: us-west-2
10        AWS_ACCESS_KEY_ID: ***
11        AWS_SECRET_ACCESS_KEY: ***
```


Links

- <https://lab.github.com/githubtraining/github-actions:-continuous-integration>
- <https://lab.github.com/githubtraining/github-actions:-publish-to-github-packages>
- <https://lab.github.com/githubtraining/github-actions:-continuous-delivery-with-aws>
- <https://help.github.com/en/actions>
- <https://github.com/marketplace?type=actions>
- <https://github.com/githubsatelliteworkshops/cd-with-actions>



Satellite