

Building and publishing a Docker image to ECR using GitHub Actions



Terence Wong

March 14, 2022 • 3 mins

GitHub Actions uses workflows so that you can include DevOps processes in any GitHub repository. GitHub Actions lets the build repository interact with various services in a deployment process. Commonly, a codebase will be built and pushed into a container registry for deployment later.

In this post, I show you how to build and push the Octopus Deploy underwater app to Amazon Elastic Container Registry (ECR) using GitHub Actions.

Prerequisites

To follow along, you need:

- An Amazon Web Services (AWS) account
- A GitHub account

This post uses the [Octopus underwater app repository](#). You can fork the repository and follow along. Alternatively, the `github-ecr` branch contains the template files you need to complete the steps in this post. You have to replace some values with your own, but I've included my values in this post as a reference.

Amazon Web Services setup

To set up AWS for GitHub Actions, you need to create an access key and an ECR repository to store the image.

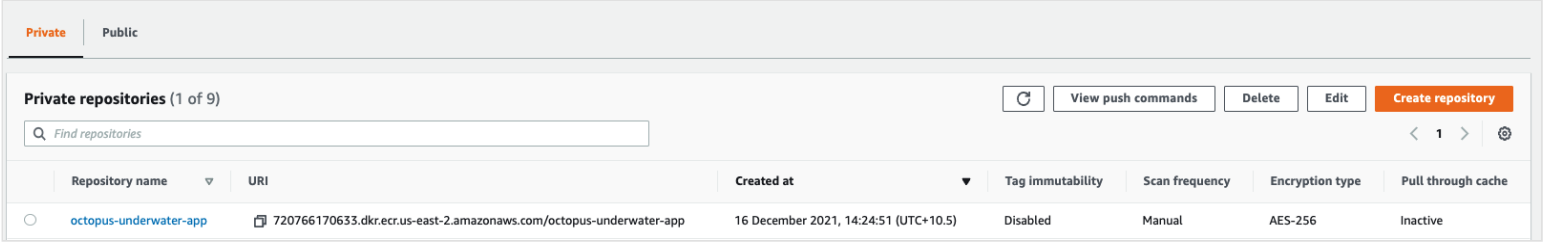
To create an access key, go to **Amazon Console**, then **IAM**, then **Users**, [\[your user\]](#), then **Security credentials**, and then **Create Access Key**.

Your browser downloads a file containing the Access Key ID and the Secret Access Key. These values are used in GitHub to authenticate to Amazon.

To create a repository, go to the **Amazon Console**, then **ECR**, and then **Create Repository**.

You need to set up an image repository for each image that you publish. Give the repository the same name you want the image to have.

You will see your repository under **Amazon ECR**, then **Repositories**. Make a note of the zone it's in, in the URI field.



GitHub setup

In this post, you build the Octopus Deploy underwater app repository and push it to Amazon ECR. You use these images in later blog posts.

Fork the repository at <https://github.com/terence-octo/octopus-underwater-app-docker>.

Go to **Settings**, then **Secrets**, and then **New repository secret**.

- **REPO_NAME** - the name of the AWS ECR repository you created
- **AWS_ACCESS_KEY_ID** - the Access Key ID from earlier
- **AWS_SECRET_ACCESS_KEY** - the Secret Access Key from earlier

You need to create a workflow file in the repository. A GitHub Actions workflow contains instructions on performing operations on the code repository. These are community-maintained steps. Several pre-built step templates allow you to do many different tasks on a code repository. In this example, you use a step template to build and push the code to an AWS ECR repository.

Create a file named `main.yml` in the `.github/workflows` directory of the root folder. Paste the following code in the `main.yml` file:

```
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

name: AWS ECR push

jobs:
  deploy:
    name: Deploy
    runs-on: ubuntu-latest

    steps:
      - name: Checkout
        uses: actions/checkout@v2

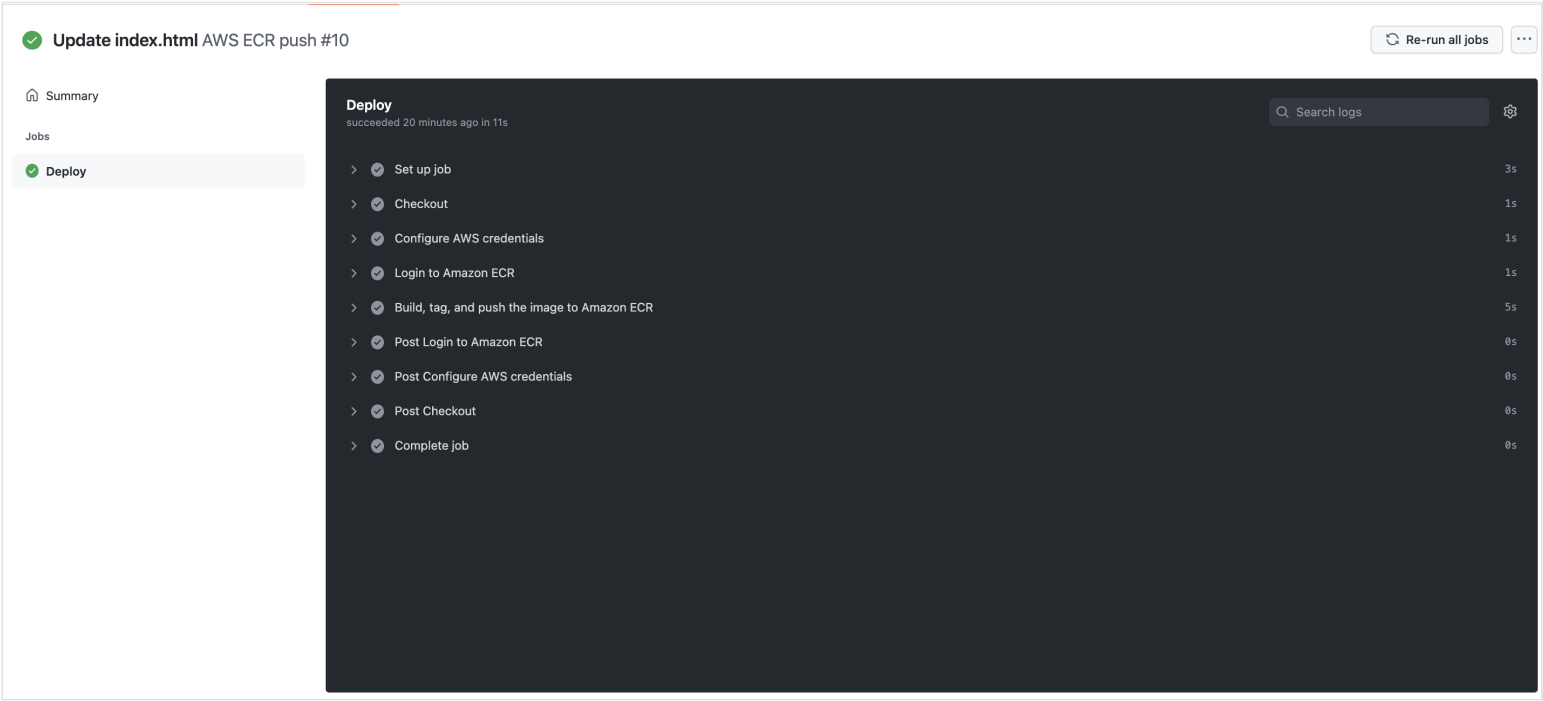
      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
          aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
          aws-region: us-east-2

      - name: Login to Amazon ECR
        id: login-ecr
        uses: aws-actions/amazon-ecr-login@v1

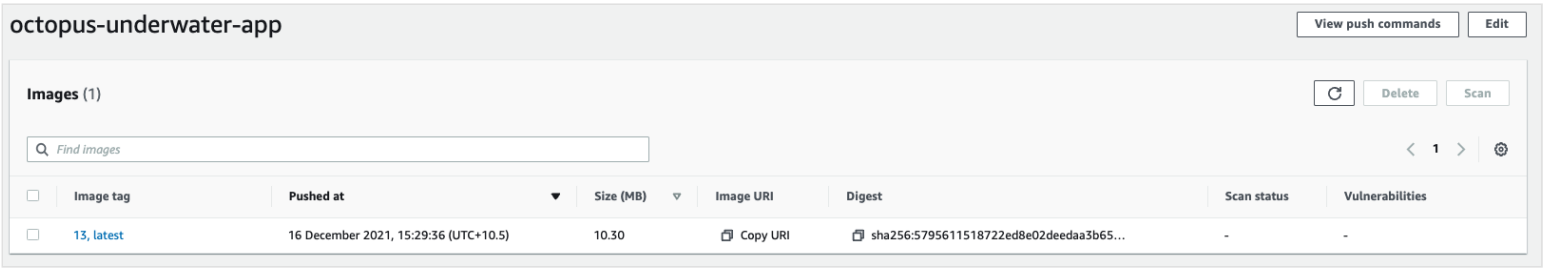
      - name: Build, tag, and push the image to Amazon ECR
        id: build-image
        env:
          ECR_REGISTRY: ${ steps.login-ecr.outputs.registry }
          ECR_REPOSITORY: ${ secrets.REPO_NAME }
          IMAGE_TAG: latest
        run: |
          # Build a docker container and push it to ECR
          docker build -t $ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG .
          echo "Pushing image to ECR..."
          docker push $ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG
          echo "::set-output name=image::$ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG"
```

GitHub starts an action by using a push or pull request on the main branch. The steps check out the code, authenticate and log into AWS, then build, tag, and push the image to Amazon ECR. A similar step template could push to other cloud repositories like Google or Microsoft.

Commit your changes, go to the **Actions** tab, and click the title of your commit message. You see the various stages of the workflow as it reaches completion.



Go to your Amazon ECR repository to view the image. Octopus Deploy can now deploy this image to a deployment target.



Conclusion

In this post, you set up a GitHub Actions workflow to build and push an image to Amazon ECR. Deployment tools like Octopus can then use this image in a later deployment stage to deploy the web application to a service. GitHub Actions allows code repositories to be part of a deployment process without additional work. GitHub Actions has several templates that developers can use to perform other deployment tasks.

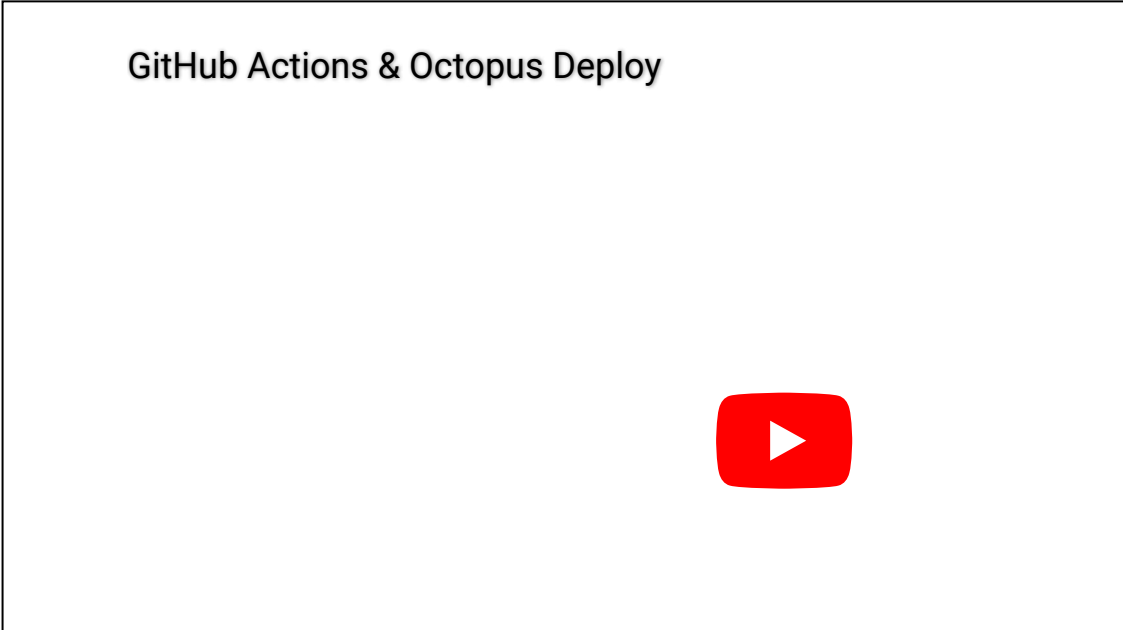
In the next post, we'll [take the image in ECR and deploy the web application](#).

Following that, we'll [use GitHub Actions and Octopus Deploy to deploy a web application to Amazon EKS](#).

[Try our free GitHub Actions workflow tool](#) to help you quickly generate customizable workflows for your GitHub Actions deployments.

You can also learn more about [building with GitHub and deploying with Octopus](#), and use our [verified actions in the GitHub Marketplace](#).

Watch our GitHub Actions webinar

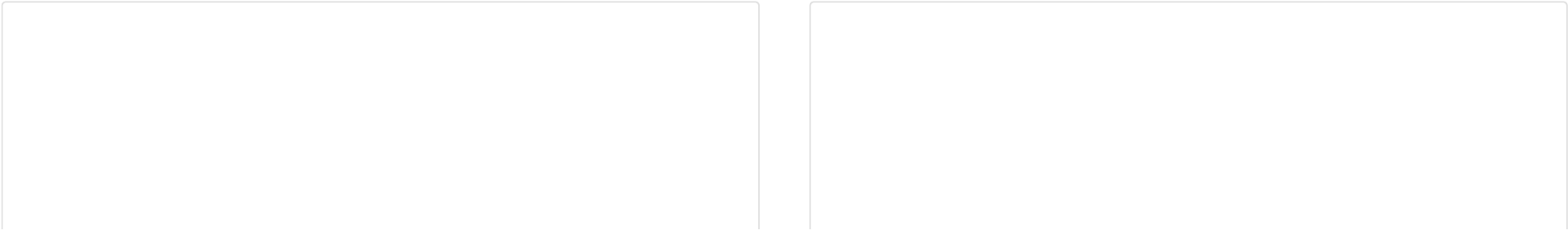


Read the rest of our [Continuous Integration series](#).

Happy deployments!

Tagged with: [DevOps](#) [CI Series](#) [Continuous Integration](#) [GitHub Actions](#)

Related posts





DevOps

Common mistakes in DevOps metrics

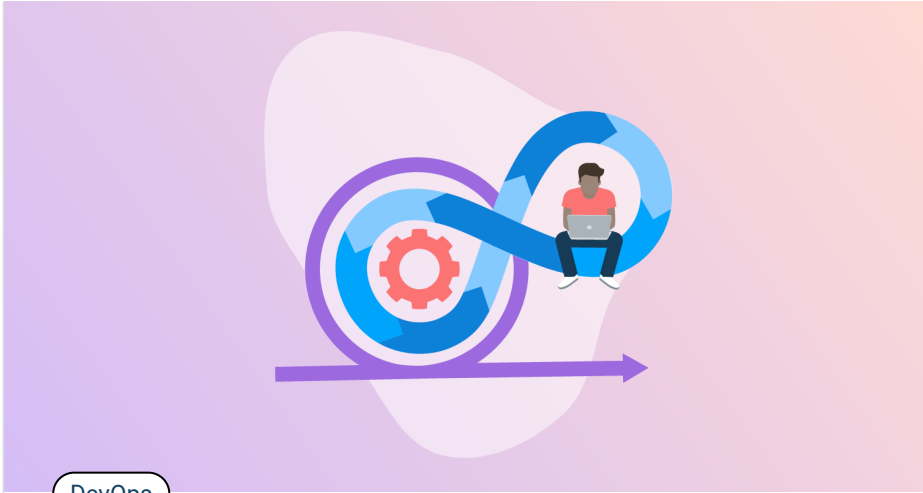


DevOps

Best practices for CI/CD



Terence Wong
November 30, 2022 • 5 mins



DevOps

Comparing Lean, Agile, and Continuous Delivery



Steve Fenton
December 5, 2022 • 6 mins

Newsletter

Logged in as Terence Wong (terence.wong@octopus.com)

Join ~48,000 DevOps professionals and sign up for the latest Octopus news, events, and opinions. No spam. Unsubscribe at any time.

Subscribe

Your privacy is important to us. Read more in our [Privacy Policy](#).

