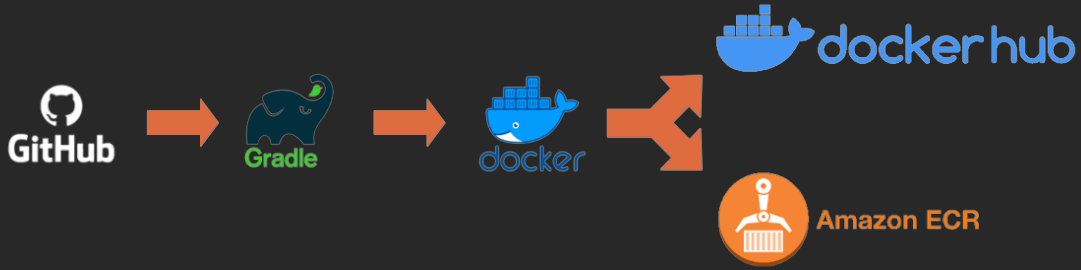
GitHub Actions, Docker, DockerHub, ECR

Dr. Isaac Griffith Idaho State University

The Gist

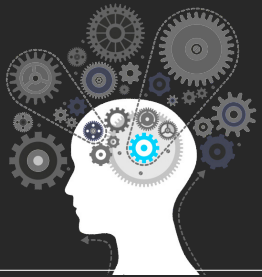


Outcomes

Idaho State University | Computer Science

After today's lecture you will be able to:

- Use GitHub Actions to
  - Automate the Build
  - Automate Creating your Docker Image
  - Automate Deploying your Docker Image to:
    - DockerHub
    - AWS ECR



GitHub Actions → DockerHub

CS 3321

GHA → DockerHub

Idaho State University | Computer Science

- **Initial Steps**
  1. Add your Docker ID as a secret to GitHub
     - Navigate to your GitHub repository and click **Settings > Secrets > New Secret**
     - Create a new secret with the name `DOCKER_HUB_USERNAME` and your Docker ID as the value
  2. Create a new Personal Access Token (PAT) on Docker Hub
     - Navigate to Docker Hub Settings and click **New Access Token**
     - Give the token a name
     - Copy the token text

GHA → DockerHub

Idaho State University | Computer Science

- **Initial Steps Continued**
  3. Add your Docker PAT as a secret to GitHub
     - Navigate to your GitHub repository and click **Settings > Secrets > New Secret**
     - Create a new secret with the name `DOCKER_HUB_ACCESS_TOKEN` and your new PAT as the value

GHA → DockerHub

Idaho State University | Computer Science

- **Setup the GitHub Actions Workflow**
  - Create a new workflow file: "`.github/workflows/push_dockerhub.yml`"

```yaml
name: Push to Docker Hub
on:
  push:
    branches:
      - 'main'

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v2

      - name: Setup JDK 17
        uses: actions/setup-java@v2
        with:
          java-version: '17'
          distribution: 'adopt'

      - name: Validate Gradle Wrapper
        uses: gradle/wrapper-validation-action@
e6e38bacfdf1a337459f332974bb2327a31aaf4b

      - name: Gradle Build
        run: ./gradlew package

      - name: Login to Docker Hub
        uses: docker/login-action@v1
        with:
          username: ${{ secrets.DOCKER_HUB_USERNAME }}
          password: ${{ secrets.DOCKER_HUB_ACCESS_TOKEN }}

      - name: Set Up Docker Buildx
        uses: docker/setup-buildx-action@v1

      - name: Docker Build and Push
        uses: docker/build-push-action@v2
        with:
          context: .
          file: ./Dockerfile
          push: true
          tags: ${{ secrets.DOCKER_HUB_USERNAME }}/project:latest
```

GHA → DockerHub

Idaho State University | Computer Science

- Push your changes
- Using the GitHub Actions web interface
  - Test your workflow
  - Tweak the workflow until it completely works

GitHub Actions → AWS ECR

CS 3321

GHA → AWS ECR

Idaho State University | Computer Science

- **Creating an ECR**
  1. Login in to the AWS Management App
  2. Navigate to the Elastic Container Registry
  3. Create a new Repository
     - Choose visibility as Private and give the repo a **Name**
     - Click create repository

GHA → AWS ECR

Idaho State University | Computer Science

- **Creating an ECR**
  4. Navigate to the AWS IAM Feature, then click **Add Role**
     - Provide a new User Name: **Github-Action-AWS-CLI-Allow-ECR** and check **Programmatic Access** which will allow the creation of a key pair.
     - Click Next: Permissions and on the next screen choose “Attach existing policies directly”.
     - In find policies search for “**AmazonEC2ContainerRegistryFullAccess” and check it
     - Click on **Next: Tags** skip and click on **Next: Review** and confirm **Create User**
  5. Once create, you need to save the Credentials File and Copy the “**Access Key ID**” and “**Secret Access Key**”

GHA → AWS ECR

Idaho State University | Computer Science

- **Connecting GHA with AWS ECR**
  1. Open your repository secret settings on GitHub (Click **Settings > Secrets**)
  2. Create the following secrets with the given values
     - Add **REPO_NAME** with the value of your ECR repo name
     - Add **AWS_ACCESS_KEY_ID** with the value of the AWS Access Key ID copied previously
     - Add **AWS_SECRET_ACCESS_KEY** with the value of the AWS Secret Access Key copied previously
     - **Note:** You can use the saved credentials file for these values

GHA → AWS ECR

Idaho State University | Computer Science

- **Setup the GitHub Actions Workflow**
  - Create a new workflow file: "`.github/workflows/push_ecr.yml`"

```yaml
name: Push to AWS ECR
on:
  push:
    branches:
      - 'main'

jobs:
  deploy:
    name: Deploy to AWS ECR
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v2
      - name: Setup JDK 17
        uses: actions/setup-java@v2
        with:
          java-version: '17'
          distribution: 'adopt'
      - name: Validate Gradle Wrapper
        uses: gradle/wrapper-validation-action@
      e6e38bacfdf1a337459f332974bb2327a31aaf4b
      - name: Gradle Build
        run: ./gradlew package
      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id:
            ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-eky:
            ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: us-west-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: 1.0
        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"
```

GHA → DockerHub

Idaho State University | Computer Science

- Push your changes
- Using the GitHub Actions web interface
  - Test your workflow
  - Tweak the workflow until it completely works
- Finally, setup a AWS Fargate Node and connect it to the ECR repository.

For Next Time

- Review this Lecture
- Come to Class

Are there any questions?