

CS423 DevOps

Assignment 3

Abdur Rahman Goraya, 2022035
Ubaid ur Rehman, 2022599

About the assignment and project:

The project was cloned from <https://github.com/eljamaki01/ReactNodeTesting>, it contained a React + Express app that was a simple shopping cart application. Very basic, but contained a front-end and back-end part with tests written for both.

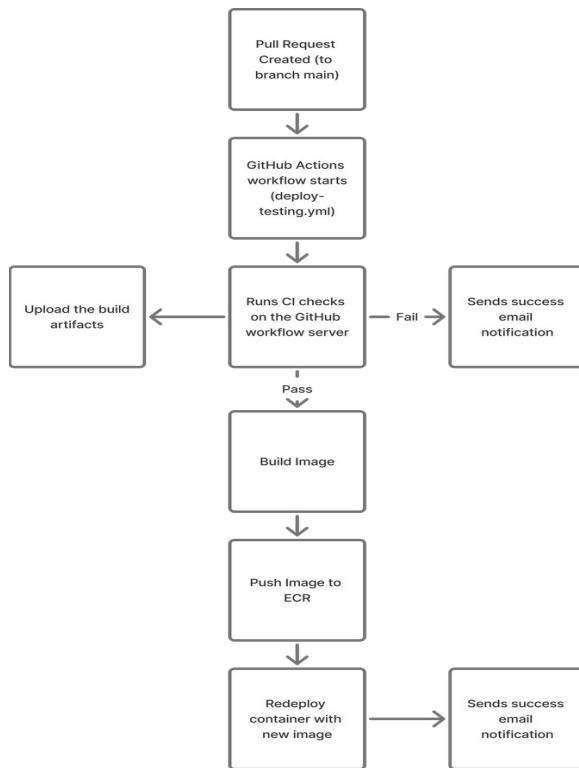
Now we set up the assignment with Docker containers, adding a dockerfile to build the images.

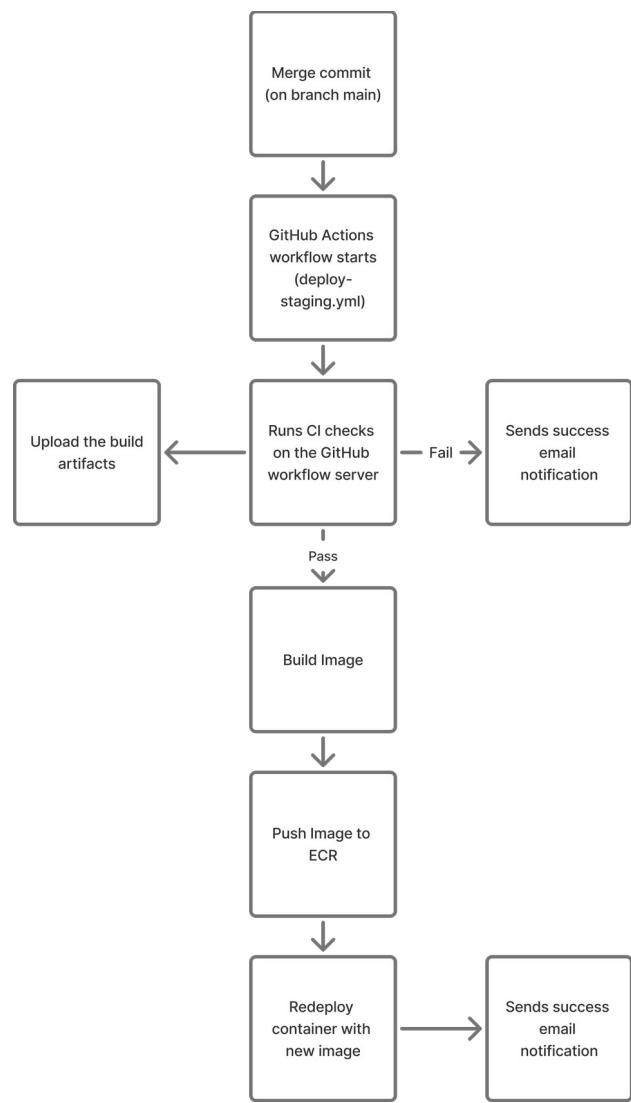
After this, I disabled both of the old GitHub Action workflows, and wrote new ones that had one major change. It builds a docker images, pushes it to ECR, then runs command on the EC2 to update the image and redeploy the container.

Difficulties we faced:

Not many difficulties were facing in this assignment. The experience from the first one carried over.

Flows:





Screenshots:

1. Creating the instances

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, 'DevOps Dev Server' is entered. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' AMI is selected. The 'Description' section notes it's a Canonical, Ubuntu, 24.04, amd64 noble image. The 'Architecture' section shows x86_64. The 'Quick Start' tab is selected. On the right, a summary shows 1 instance, the Canonical, Ubuntu, 24.04 AMI, t2.micro instance type, devops-sg security group, and 1 volume(s) - 8 GiB storage. A note about the free tier is displayed. At the bottom are 'Cancel', 'Launch instance', and 'Preview code' buttons.

2. Creating the IAM role for EC2 instances

The screenshot shows the 'Role DevOps created' confirmation page. It displays the 'DevOps' role's summary, including its ARN (arn:aws:iam::615299766355:role/DevOps), creation date (December 21, 2025), and maximum session duration (1 hour). The 'Permissions' tab is active, showing one managed policy: 'AmazonEC2ContainerRegistryPowerUser'. The 'Permissions boundary' and 'Generate policy based on CloudTrail events' sections are also visible.

3. Assigning the IAM role

The screenshot shows the AWS IAM Modify IAM role page for an EC2 instance. The instance ID is i-0a880757348dfb166 (Devops Staging Server). The 'IAM role' dropdown menu is open, showing 'DevOps' selected. Other options include 'Create new IAM role'. At the bottom right are 'Cancel' and 'Update IAM role' buttons.

4. Setting up instances

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

System load: 0.15      Processes:          109
Usage of /: 29.7% of 6.71GB Users logged in:    0
Memory usage: 25%      IPv4 address for enX0: 172.31.29.30
Swap usage: 0%
```

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

```
Last login: Sun Dec 21 17:33:51 2025 from 154.196.93.59
ubuntu@ip-172-31-29-30:~$ curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh | bash
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 16631 100 16631 0 0 417K 0:--:-- --:-- --:--:-- 427K
=> nvm is already installed in /home/ubuntu/.nvm, trying to update using git
=> Compressing and cleaning up git repository

=> nvm source string already in /home/ubuntu/.bashrc
=> bash_completion source string already in /home/ubuntu/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
ubuntu@ip-172-31-29-30:~$ sudo apt install libatomic1
Reading package lists...
Building dependency tree...
Reading state information...
libatomic1 is already the newest version (14.2.0-4ubuntu2-24.04).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-29-30:~$
```

4. Create GitHub Actions User and then access token

The screenshot shows the 'Review and create' step of the AWS IAM 'Create user' wizard. The process is at Step 3: Set permissions. The user details section shows a User name of 'GitHub-Actions' and a Console password type of 'None'. The Permissions summary table lists a single permission: 'AmazonEC2ContainerRegistryFullAccess' (AWS managed, Permissions policy). The Tags section is empty. At the bottom, there are 'Cancel', 'Previous', and 'Create user' buttons.

The screenshot shows the 'Retrieve access keys' step of the AWS IAM 'Create access key' wizard. The process is at Step 3: Retrieve access keys. A green banner at the top states: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' The Access key section displays the Access key 'AKIAY6QVZLBQ7R6LEVE' and the Secret access key, which is shown as a series of asterisks. The Access key best practices section lists several guidelines. At the bottom, there are 'Download .csv file' and 'Done' buttons.

5. Setting up the new secrets and variables

The screenshot shows the GitHub Actions secrets page for a repository named 'flankedgonerogue/devops-assignment'. The left sidebar includes sections for Environments, Codespaces, Pages, Security, Advanced Security, Deploy keys, Secrets and variables, Actions, Codespaces, Dependabot, Integrations, GitHub Apps, Email notifications, and Autolink references. The main area displays a table titled 'Repository secrets' with the following data:

| Name | Last updated |
|------------------------|---------------|
| APP_DIRECTORY | 2 hours ago |
| AWS_ACCESS_KEY_ID | 3 minutes ago |
| AWS_REGION | 9 minutes ago |
| AWS_SECRET_ACCESS_KEY | 3 minutes ago |
| AWS_SSH_PRIVATE_KEY | 3 hours ago |
| AWS_SSH_USERNAME | 3 hours ago |
| ECR_REPOSITORY_STAGING | 3 minutes ago |
| ECR_REPOSITORY_TESTING | 3 minutes ago |
| QA_EMAIL | 2 hours ago |
| SMTP_FROM_EMAIL | 2 hours ago |
| SMTP_PASSWORD | 2 hours ago |
| SMTP_PORT | 2 hours ago |
| SMTP_SERVER | 2 hours ago |
| SMTP_USERNAME | 2 hours ago |
| STAGING_SERVER_IP | 2 hours ago |
| TESTING_SERVER_IP | 2 hours ago |

At the bottom of the page, there is a footer with links to GitHub, Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and a link to not share personal information.

5. Viewing the success e-mails

The screenshot shows the Ethereal Email interface. The top navigation bar includes tabs for Assignment 4.pdf, Ethereal Email, React App, and a search bar. Below the navigation is a header with the Ethereal logo, Home, FAQ, Help, Messages, and Logout links. The main content area displays an email message with the following details:

Public URL of this message
Subject: Testing Deployment Successful - flankedgonerogue/devops-assignment
From: <derrel.rau51@ethereal.email>
To: <flankedgonerogue@gmail.com>
Time: Today at 23:20
Message-ID: <35df14ae-14b7-c033-4ee8-34c58dc56ddf@ethereal.email>

The message content is as follows:

Testing Server Deployment Successful
Please check it out at <http://18.136.211.207>.

Deployment Details
Repository: flankedgonerogue/devops-assignment Branch: dev Pull Request: #2 PR Title: PR #2 Triggered By: flankedgonerogue Image: 615299766355.dkr.ecr.ap-southeast-1.amazonaws.com/react-node-app-testing:907d4e28e70c068cdc17e79a6175876342e3e20

At the bottom of the interface, there is a footer with the text: © 2017—2022 Ethereal Email info@ethereal.email | service provided by EmailEngine.