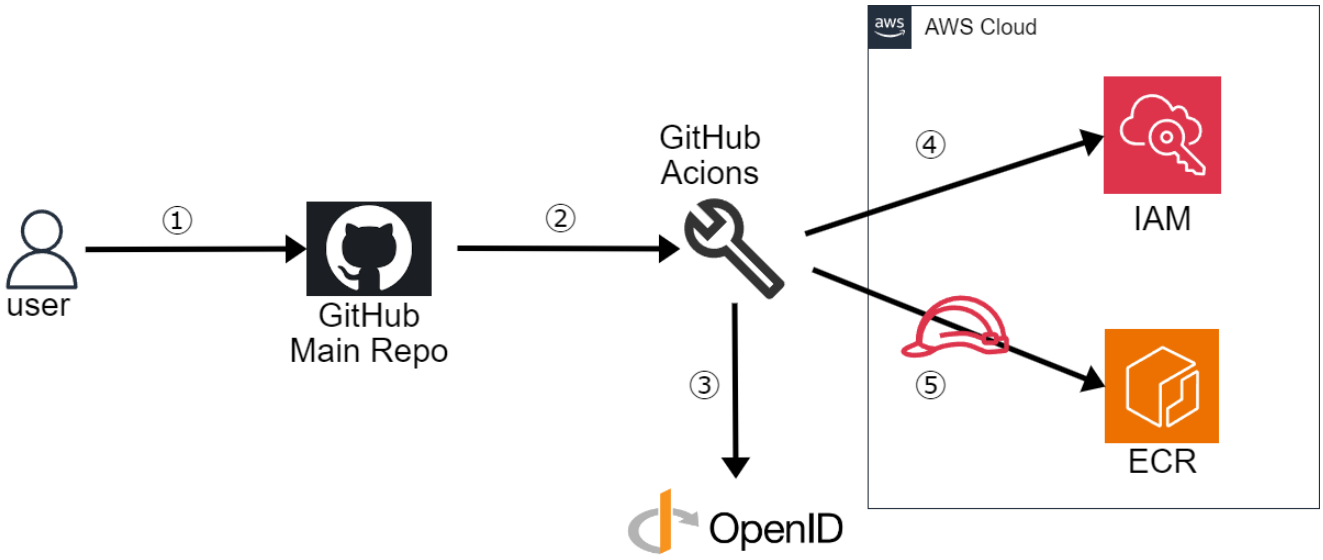
概要

GitHubリポジトリでDockerコンテナをビルドするDockerfileを作成すると同時に、Dockerイメージが自動でECRにアップロードされるようにする



1. ユーザはGitHubにイメージをPush、
2. main branch の更新をトリガーとしてGithub Actions が動作
3. GitHub が OIDC に Tokenリクエスト
4. OIDCトークンで IAMに一時クレデンシャルを リクエスト
5. 一時資格情報よりIAMロールでECRにイメージをPUT

ポイント

OIDC認証を利用（ **アクセスID、シークレットアクセスキーを使わない** ）

実装の流れ

1. GitHub OIDC プロバイダーをAWSアカウント側で準備
2. 信頼関係を登録したGitHubリポジトリ用のIAM Roleを用意
3. GitHub Actions のワークフローを作成

OIDC プロバイダー作成

Configuring OpenID Connect in Amazon Web Services[1]
GitHub Actions で OIDC を使用して AWS 認証を行う[2]

OIDC認証用IAMロール作成

カスタム信頼ポリシーを作成（公式サンプルをベース）

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "Federated": "arn:aws:iam::000123456789:oidc-provider/token.actions.githubusercontent.com"
         },
         "Action": "sts:AssumeRoleWithWebIdentity",
         "Condition": {
            "StringEquals": {
               "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
               "token.actions.githubusercontent.com:sub": "repo:Test-User/techblue-dev:ref:refs/heads/main"
            }
         }
      }
   ]
}
```

インラインポリシーを追加（公式サンプル[3]をベース）

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "ecr:GetAuthorizationToken",
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Action": [
            "ecr:CompleteLayerUpload",
            "ecr:UploadLayerPart",
            "ecr:InitiateLayerUpload",
            "ecr:BatchCheckLayerAvailability",
            "ecr:PutImage"
         ],
         "Resource": "arn:aws:ecr:ap-northeast-1:000123456789:repository/techblue-dev"
      }
   ]
}
```

GitHub Actionsワークフロー作成

テスト用のDockerファイル作成

Dockerfile

```
FROM alpine:3
```

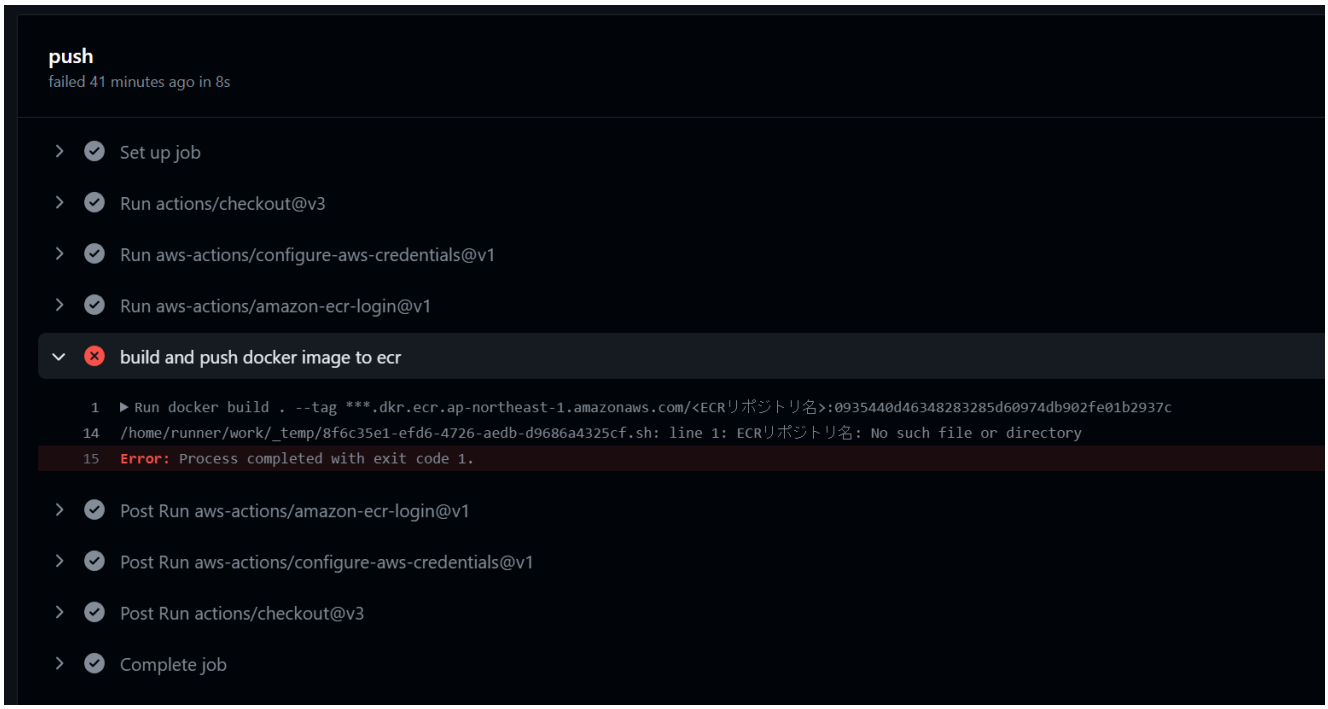
ワークフロー作成

```
name: ecr push image
on:
  push:
jobs:
  push:
    runs-on: ubuntu-latest
    # `permissions` を設定しないと OIDC が使えないので注意
    permissions:
      id-token: write
      contents: read
    steps:
      - uses: actions/checkout@v3
      # AWS 認証
      - uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-region: "ap-northeast-1"
          role-to-assume: "arn:aws:iam::767397806995:role/Githubacitions-to-ECR-role"
      # ECR ログイン
      - uses: aws-actions/amazon-ecr-login@v1
        id: login-ecr # outputs で参照するために id を設定
      # Docker イメージを build・push する
      - name: build and push docker image to ecr
        env:
          # ECR レジストリを `aws-actions/amazon-ecr-login` アクションの `outputs.registry` から取得
          REGISTRY: ${{ steps.login-ecr.outputs.registry }}
          # イメージを push する ECR リポジトリ名
          REPOSITORY: "techblue-dev"
          # 任意のイメージタグ
          # 今回は Git のコミットハッシュにしておく
          IMAGE_TAG: ${{ github.sha }}
        run: |
          docker build . --tag ${{ env.REGISTRY }}/${{ env.REPOSITORY }}:${{ env.IMAGE_TAG }}
          docker push ${{ env.REGISTRY }}/${{ env.REPOSITORY }}:${{ env.IMAGE_TAG }}
```

エラー発生時の確認

1. GitHub Actionsでエラー箇所を確認
2. Clouldtailでログを確認

GitHub Actionsでエラー箇所を確認



よく出るやつ（Not authorized to perform sts:AssumeRoleWithWebIdentity）

```
Run aws-actions/configure-aws-credentials@v1
  with:
    aws-region: ap-northeast-1
    role-to-assume: arn:aws:iam::000123456789:role/Githubacitions-to-ECR-role
    audience: sts.amazonaws.com
Error: Not authorized to perform sts:AssumeRoleWithWebIdentity
```

Cloudtailでエラーログを確認

JSON形式でログがS3に格納される。

```
  {
    "eventVersion": "1.08",
    "userIdentity": {
      "type": "WebIdentityUser",
      "principalId": "arn:aws:iam::000123456789:oidc-
provider/token.actions.githubusercontent.com:sts.amazonaws.com:repo:Test-User/techblue-
dev:ref:refs/heads/main",
      "userName": "repo:Test-User/techblue-dev:ref:refs/heads/main",
      "identityProvider": "arn:aws:iam::000123456789:oidc-provider/token.actions.githubusercontent.com"
    },
    "eventTime": "2024-03-17T11:33:52Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRoleWithWebIdentity",
    "awsRegion": "ap-northeast-1",
    "sourceIPAddress": "20.42.16.112",
    "userAgent": "aws-sdk-nodejs/2.1112.0 linux/v16.20.2 configure-aws-credentials-for-github-actions
promise",
    "errorCode": "AccessDenied",          <-- エラー箇所
    "errorMessage": "An unknown error occurred",
    "requestParameters": {
      "roleArn": "arn:aws:iam::000123456789:role/Githubacitions-to-ECR-role",
      "roleSessionName": "GitHubActions",
      "durationSeconds": 3600
    },
    "responseElements": null,
    "requestID": "01733dfa-d283-4dbc-8241-e6107bb3b0fe",
    "eventID": "554ffdae-82d0-4512-b08c-1a9b15bdc264",
    "readOnly": true,
    "resources": [
      {
        "accountId": "000123456789",
        "type": "AWS::IAM::Role",
        "ARN": "arn:aws:iam::000123456789:role/Githubacitions-to-ECR-role"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "000123456789",
    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_128_GCM_SHA256",
      "clientProvidedHostHeader": "sts.ap-northeast-1.amazonaws.com"
    }
  },
```

参考

GitHub ActionsとAWSの連携におけるOIDC認証の活用[4]
GitHub Actions から ECR に Docker イメージを push する[5]
アマゾン ウェブ サービスでの OpenID Connect の構成[6]
Amazon Elastic Container Serviceへのデプロイ[7]

リンク

[1] Configuring OpenID Connect in Amazon Web Services, https://docs.github.com/en/actions/deployment/security-hardening-your-deployments/configuring-openid-connect-in-amazon-web-services
[2] GitHub Actions で OIDC を使用して AWS 認証を行う, https://zenn.dev/kou_pg_0131/articles/gh-actions-oidc-aws
[3] 公式サンプル, https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/image-push.html
[4] GitHub ActionsとAWSの連携におけるOIDC認証の活用, https://qiita.com/satooshi/items/0c2f5a0e2b64a1d9a4b3
[5] GitHub Actions から ECR に Docker イメージを push する, https://zenn.dev/kou_pg_0131/articles/gh-actions-ecr-push-image#1.-github-actions-用の-id-プロバイダと-iam-ロールを作成
[6] アマゾン ウェブ サービスでの OpenID Connect の構成, https://docs.github.com/ja/actions/deployment/security-hardening-your-deployments/configuring-openid-connect-in-amazon-web-services
[7] Amazon Elastic Container Serviceへのデプロイ, https://docs.github.com/ja/actions/deployment/deploying-to-your-cloud-provider/deploying-to-amazon-elastic-container-service