GitHub Actions and Flux cd

Continuous Integrations & Continuous Deployment using the respective tools GitHub Actions & FluxCD for lugbee project **TESTING**

GitHub Actions & FluxCD

GitHub Actions

Automation of our software workflows, now with CI/CD. Build, test, and deploy code right from GitHub ***GitHub Actions are event-driven***

Reference GitHub Actions docs

Flux CD

Flux is based on a set of Kubernetes API extensions (“custom resources”), which control how git repositories and other sources of configuration are applied into the cluster (“synced”).

Reference Flux docs

> ⚠ This implementation is in the Testing On this Repos lb-booking, lb-file-service, lb-search-service
>
> Team DevOps testing the GitHub Actions and Flux CD

GitHub Actions Setup **TESTING**

Environment variable (Global)

Default variables GitHub Provides

```
env:
  GITHUB_SHA: ${{ github.sha }}
  GITHUB_REF: ${{ github.ref }}
  REGISTRY_HOSTNAME: gcr.io
  PROJECT_NAME: ${{ secrets.PROJECT_ID }}
  PROJECT_NAMESPACE: lb-booking
  IMAGE_NAME: lbms
  SLACK_WEBHOOK_URL: ${{ secrets.SLACK_WEBHOOK_URL }}
```

Jobs in GitHub Actions

A job is a set of steps that execute on the same runner. By default, a workflow with multiple jobs will run those jobs in parallel. You can also configure a workflow to run jobs sequentially.

`job: setup-build-publish`

Steps are used in yml

A step is an individual task that can run commands in a job. A step can be either an action or a shell command. Each step in a job executes on the same runner, allowing the actions in that job to share data with each other.

Steps are Used (workflow)

act10ns/slack@v1

Check Out Repo

Set up JDK 11

Check Java Version

Authenticate Docker

Build Docker Image

Publish Docker Image

Bump Helm Chart Version

Commit & Push changes

Slack Notifications

> Checkout@v2 is the feature of GitHub for (Clone the repo to The Choice of your OS here we have the ubuntu-latest)

```
name: Lugbee CI/CD

on:
  create:
    tags:
      - v*
env:
  GITHUB_SHA: ${{ github.sha }}
  GITHUB_REF: ${{ github.ref }}
  REGISTRY_HOSTNAME: gcr.io
  PROJECT_NAME: ${{ secrets.PROJECT_ID }}
  PROJECT_NAMESPACE: lb-booking
  IMAGE_NAME: lbms
  SLACK_WEBHOOK_URL: ${{ secrets.SLACK_WEBHOOK_URL }}

jobs:
  setup-build-publish:
    name: Setup, Build, and Publish
    runs-on: ubuntu-latest
    permissions:
      contents: read
      packages: write
    steps:
    - uses: act10ns/slack@v1
      with:
        status: starting
        channel: '#lugbee-actions'
      if: always()

    - name: Check Out Repo
      uses: actions/checkout@v2
      with:
        token: ${{ secrets.GIT_TOKEN }}
        ref: master

    - name: Set up JDK 11
```

```
      uses: actions/setup-java@v2
      with:
        java-version: '11'
        distribution: 'adopt'

    - name: Check Java Version
      run: |
        java --version
    - name: Build with Maven
      run: |
        mvn -B package --file pom.xml && cp target/*.jar "$IMAGE_NAME".
jar
    - name: Setup Google Cloud CLI
      uses: google-github-actions/setup-gcloud@master
      with:
        project_id: ${{ secrets.PROJECT_ID }}
        service_account_key: ${{ secrets.GCR_KEY }}
        export_default_credentials: false

    - name: Authenticate Docker
      run: |
        gcloud auth configure-docker
    - name: Build Docker Image
      run: |
        sudo apt install grep sed
        DOCKER_BUILDKIT=1 docker build -t "$REGISTRY_HOSTNAME"
/"$PROJECT_NAME"/"$PROJECT_NAMESPACE"/"$IMAGE_NAME":main-"$(echo
${GITHUB_SHA} | cut -c1-7)" .
        cat ./charts/dev/$IMAGE_NAME/values.yaml | grep "tag" | awk
'{print $2}' > ~/.current_tag
        export CURRENT_IMAGE_TAG="$(cat ~/.current_tag)"
        echo $CURRENT_IMAGE_TAG
        docker images | awk 'NR >= 2 && NR <=2 {print $2}' > ~/.
latest_tag
        export LATEST_IMAGE_TAG="$(cat ~/.latest_tag)"
        echo $LATEST_IMAGE_TAG
        echo "commit_tag=$LATEST_IMAGE_TAG" >> $GITHUB_ENV
        sed -i -e "s/tag: $CURRENT_IMAGE_TAG/tag: $LATEST_IMAGE_TAG/" .
/charts/dev/$IMAGE_NAME/values.yaml
        sed -i "s/^appVersion:.*$/appVersion: $LATEST_IMAGE_TAG/" .
/charts/dev/$IMAGE_NAME/Chart.yaml
        cat ./charts/dev/$IMAGE_NAME/Chart.yaml | grep "appVersion"
        cat ./charts/dev/$IMAGE_NAME/values.yaml | grep "tag"
        rm "$IMAGE_NAME".jar
    - name: Publish Docker Image
      run: |
        docker tag "$REGISTRY_HOSTNAME"/"$PROJECT_NAME"
/"$PROJECT_NAMESPACE"/"$IMAGE_NAME":main-"$(echo ${GITHUB_SHA} | cut -
c1-7)" "$REGISTRY_HOSTNAME"/"$PROJECT_NAME"/"$PROJECT_NAMESPACE"
/"$IMAGE_NAME":main-"$(echo ${GITHUB_SHA} | cut -c1-7)"
```

```
        docker push "$REGISTRY_HOSTNAME"/"$PROJECT_NAME"/"$PROJECT_NAMESPACE"/"$IMAGE_NAME":main-"$(echo ${GITHUB_SHA} | cut -c1-7)"
    - name: Helm Reader
      id: chart
      uses: jacobtomlinson/gha-read-helm-chart@master
      with:
        path: ./charts/dev/lbms/

    - name: Bump Helm Chart Version
      run: |
        echo "Name - ${{ steps.chart.outputs.name }}"
        echo "Version - ${{ steps.chart.outputs.version }}"
        echo "App Version - ${{ steps.chart.outputs.appVersion }}"
        echo ${{ steps.chart.outputs.version }} | ( IFS=".$IFS" ; read a b c && echo $a.$b.$((c + 1)) ) > ~/.helm_ver
        export HELM_VER="$(cat ~/.helm_ver)"
        echo $HELM_VER
        sed -i "s/^version:.*$/version: $HELM_VER/" ./charts/dev/$IMAGE_NAME/Chart.yaml
    - name: Commit & Push changes
      run: |
        git config user.name lugbee-gitops
        git config user.email gitops@github.com
        git add .
        git commit -m 'Updated Charts to ${{ env.commit_tag }}'
        git push
    - name: Slack Notifications
      uses: act10ns/slack@v1
      with:
        status: ${{ job.status }}
        steps: ${{ toJson(steps) }}
        channel: '#lugbee-actions'
      if: always()
```

Flux CD

⚠ Docs will be publish soon after alterations