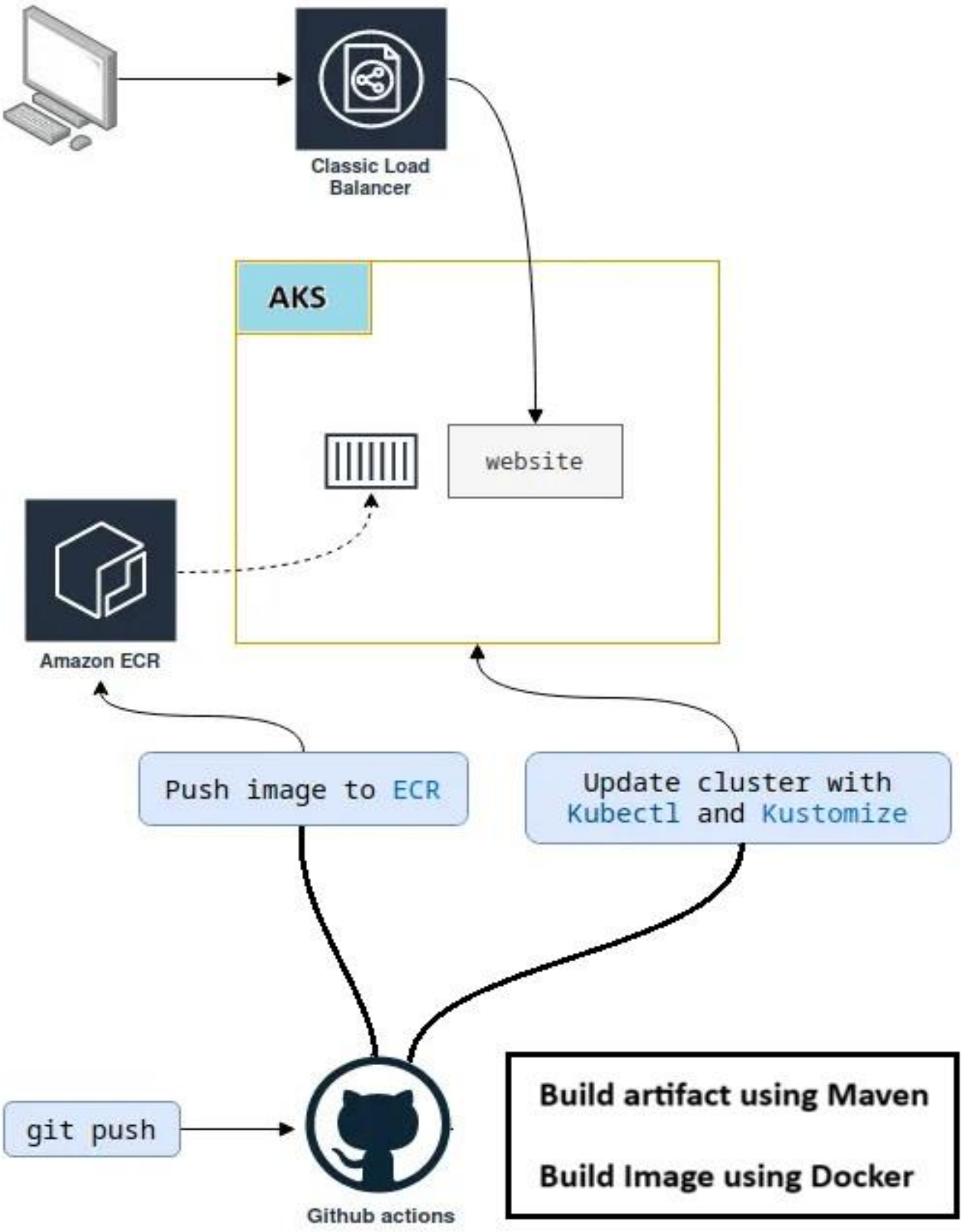
# Using github-action deploy your app to AKS ( using ECR as repository )



Steps:

1. Make your aks cluster
2. Write work flow
   - Step 1. Install java
   - Step 2. Build with maven
   - Step 3. Upload build artifact
   - Step 4. List build directory contents
   - Step 5. Configure Aws credentials
   - Step 6. Login in to amazon ECR
   - Step 7. Build docker image
   - Step 8. Push docker image to ECR
   - Step 9. Azure login
   - Step 10. Set up kubelogin for non-interactive login
   - Step 11. Get k8s context
   - Step 12. Interact and deploy to EKS

**IMPORTANT STEPS:**

**1. Aws configure :**

- Add aws credentials into repository secret
  - AWS_ACCESS_KEY_ID
  - AWS_SECRET_ACCESS_KEY

**2. Azuere login**

- To log in to Azure using GitHub Actions, you can follow these steps:

1. **Create a Service Principal:**
   - Use the Azure CLI to create a Service Principal with the necessary permissions. Run the following command:

     **az ad sp create-for-rbac --name "myApp" --role contributor --scopes /subscriptions/<subscription-id> --sdk-auth**

   - This command will output a JSON object containing your credentials. Copy this JSON object.
2. **Add the Credentials to GitHub Secrets:**
   - Go to your GitHub repository, navigate to **Settings** > **Secrets and variables** > **Actions**.
   - Add a new secret with the name **AZURE_CREDENTIALS** and paste the JSON object you copied earlier.
3. **Create a GitHub Actions Workflow:**

- In your repository, create a .github/workflows/azure-login.yml file with the following content:
- - name: Azure Login
- uses: azure/login@v1
- with:
- creds: ${{ secrets.AZURE_CREDENTIALS }}

3.for deploying a deployment in aks using an image from ecr.

- aws configure
- make a secret for ecr-to-eks

**$ kubectl create secret docker-registry ecr-registry-secret --docker-server=559050231342.dkr.ecr.ap-south-1.amazonaws.com --docker-username=AWS --docker-password=$(aws ecr get-login-password)**

- **Use the secret into your deployment**

**Deployment file:**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: regapp-deployment
  labels:
     app: regapp

spec:
  replicas: 2
  selector:
    matchLabels:
      app: regapp

  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
      - name: regapp
        image: 559050231342.dkr.ecr.ap-south-1.amazonaws.com/github-repo:latest
        imagePullPolicy: Always
        ports:
        - containerPort: 8080

      imagePullSecrets:
      - name: ecr-registry-secret
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
```

**Work flow flow file:**

```yaml
name: Java CI with Maven

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

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v4
    - name: Set up JDK 17
      uses: actions/setup-java@v4
      with:
        java-version: '17'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn -B package --file pom.xml

    - name: Upload build artifact
      uses: actions/upload-artifact@v4
      with:
        name: war-file
        path: /home/runner/work/java-pr/java-pr/webapp/target/*.war

    - name: List build directory contents
      run: ls -la /home/runner/work/java-pr/java-pr/webapp/target/

    - name: Configure AWS credentials
```

```yaml
    - 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: ap-south-1
    - name: Log in to Amazon ECR
      id: login-ecr
      uses: aws-actions/amazon-ecr-login@v1

    - name: Build Docker image
      run: |
        docker build -t github-repo .
        docker tag github-repo:latest 559050231342.dkr.ecr.ap-south-1.amazonaws.com/github-repo:latest
    - name: Push Docker image to ECR
      run: |
        docker push 559050231342.dkr.ecr.ap-south-1.amazonaws.com/github-repo:latest

    - name: Azure Login
      uses: azure/login@v1
      with:
        creds: ${{ secrets.AZURE_CREDENTIALS }}

    - name: Azure CLI script
      run: |
        az account show

    - name: Set up kubelogin for non-interactive login
      uses: azure/use-kubelogin@v1
      with:
        kubelogin-version: 'v0.0.25'



      # Retrieves your Azure Kubernetes Service cluster's kubeconfig file
    - name: Get K8s context
      uses: azure/aks-set-context@v3
      with:
        resource-group: dev-grp
        cluster-name: aa-cluster
        admin: 'false'
        use-kubelogin: 'true'

    - name: Deploy to EKS
      env:
        ECR_REGISTRY: ${{ steps.login-ecr.outputs.registry }}
        ECR_REPOSITORY: github-repo
        IMAGE_TAG: ${{ github.sha }}
      run: |
        kubectl rollout restart deployment/regapp-deployment

```

Docker file:

```
1    FROM tomcat:latest
2    RUN cp -R  /usr/local/tomcat/webapps.dist/*  /usr/local/tomcat/webapps
3    COPY webapp/target/*.war /usr/local/tomcat/webapps
4    RUN ls -la /usr/local/tomcat/webapps
```

---