**NODE CI TEMPLATE**

#this is the CI yaml file for the project which is based on the node.js

name: CI

on:

push

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout

uses: actions/checkout@v3.3.0

#setting up the node.js environmentfor the project

- name: Setup Node.js environment

uses: actions/setup-node@v3.6.0

with:

node-version: ${{env.NODE\_VERSION}}

#giving the commands for the npm to install all the dependencies

- name: Install NPM dependencies

run: ${{secrets.DEPENDENCY\_COMMANDS}}

#giving the commands for the npm to run the test

- name: run the test

run: ${{secrets.RUN\_TEST\_COMMANDS}}

#uploading the files or the directory to the github server for which it can be used in other workflows for further usage

- name: Upload a Build Artifact

uses: actions/upload-artifact@v3.1.2

with:

#specify the name of the folder in which the artifact is uploaded

name: ${{secrets.DESTINATION\_FOLDER\_NAME}}

#specify the path from where that the artifact has to be upload to the github server.

path: ${{secrets.ARTIFACT\_SOURCE\_PATH}}

**DOTNET CI TEMPLATE**

name: DOTNET CI TEMPLATE

on: [push]

jobs:

build:

runs-on: windows-2019

steps:

- name: Checkout repository

uses: actions/checkout@v2

- name: Use .NET Core sdk 2.x

uses: actions/setup-dotnet@v1

with:

dotnet-version: '6.0.x'

- name: Run the build command

run: |

<RUN\_THE\_BUILD\_COMMAND>

- name: Upload the artifact

uses: actions/upload-artifact@v2

with:

name: <NAME\_OF\_THE\_ARTIFACT>

path: <PATH\_TO\_THE\_ARTIFACT>

**MAVEN CI TEMPLATE**

name: Java CI with Maven

on:

push:

branches: [ main ]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Set up JDK 11

uses: actions/setup-java@v2

with:

#specify the java version thet nedd to be set

java-version: ${{env.JAVA\_VERSION}}

#giving the commands for the maven build

- name: Build with Maven

run: ${{secrets.MAVEN\_BUILD\_COMMANDS}}

#giving the commands for the maven test

- name: Test with Maven

run: ${{secrets.MAVEN\_TEST\_COMMANDS}}

#uploading the files or the directory to the github server for which it can be used in other workflows for further usage

- name: Upload a Build Artifact

uses: actions/upload-artifact@v3.1.2

with:

#specify the name of the folder in which the artifact is uploaded

name: ${{secrets.DESTINATION\_FOLDER\_NAME}}

#specify the path from where that the artifact has to be upload to the github server.

path: ${{secrets.ARTIFACT\_SOURCE\_PATH}}

**DOCKER CI TEMPLATE**

name: Docker Compose

on:

push:

branches: [ main ]

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v2

- name: Install Docker

uses: docker/setup-docker@v1

- name: Build and run Docker Compose

run: |

docker-compose -f docker\_composed.yml up –build

**SONARQUBE CI TEMPLATE**

name: Node.js CI

on:

push:

branches:

- main

env:

SONAR\_HOST\_URL: <sonarqube-url>

SONAR\_TOKEN: <sonar-token>

jobs:

build-and-deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Use Node.js

uses: actions/setup-node@v1

with:

node-version: '14.x'

- name: Install dependencies

run: npm install

- name: Build and package app

run: npm run build

- name: Run tests with SonarQube

uses: sonarsource/sonarqube-scan-action@v1

env:

SONAR\_HOST\_URL: ${{ env.SONAR\_HOST\_URL }}

SONAR\_TOKEN: ${{ env.SONAR\_TOKEN }}

**OWASP ZAP CI TEMPLATE**

name: Node.js CI

on:

push:

branches:

- main

jobs:

build-and-deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Use Node.js

uses: actions/setup-node@v1

with:

node-version: '14.x'

- name: Install dependencies

run: npm install

- name: ZAP Scan

uses: zaproxy/action-full-scan@v0.4.0

with:

token: ${{ secrets.GITHUB\_TOKEN }}

docker\_name: <DOCKER\_NAME>

target: <TARGET\_URL>

rules\_file\_name: <PATH>

cmd\_options: '-a'

- name: Build and package app

run: npm run build

**AZURE APP SERVICES CD TEMPLATE**

name: Deploy to Azure App Service

on: workflow\_dispatch

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- name: Download artifact

uses: actions/download-artifact@v2

with:

#specify the name of the artifact in which the artifact in the CI pipeline is uploaded

name: ${{env.DESTINATION FOLDER\_NAME}}

- name: Azure login

uses: azure/login@v1

with:

creds: ${{ secrets.AZURE\_CREDENTIALS }}

- name: Deploy to Azure App Service

uses: azure/webapps-deploy@v2

with:

#specify the name of the App that it was creted in the azure app services

app-name: ${{ secrets.PROD\_APP\_NAME }}

#specify the name of the deployment slot in which the deployment has to be done

slot-name: ${{ env.DEPLOYMENT\_SLOT\_NAME }}

#specify the package i.e. the artifact path

package: ${{ github.workspace }}/app-${{ github.run\_number }}.zip

#specify the publish profile which was created in the azure web app services

publish-profile: ${{ secrets.PROD\_PUBLISH\_PROFILE }}

**AZURE VM CD TEMPLATE**

name: Deploy to Azure VM

on: workflow\_dispatch

deploy:

#for this we need to add the VM as the self-hosted runner in that repository settings

runs-on: [self-hosted]

steps:

- name: run the commands to remove the old artifact

run: <RUN\_THE\_COMMANDS>

- name: Download artifact from build job

uses: actions/download-artifact@v3

with:

name: <ARTIFACT\_NAME>  
 path: <PATH\_TO\_DOWNLOAD>

**AZURE FUNCTION APP CD TEMPLATE**

name: AZURE FUNCTION APP CD TEMPLATE

on: [push]

jobs:

build:

runs-on: windows-2019

steps:

- name: Download artifact from build job

uses: actions/download-artifact@v3

with:

name: <ARTIFACT\_NAME>

path: <PATH\_TO\_DOWNLOAD>

- name: Azure login

uses: azure/login@v1

with:

creds: ${{ secrets.AZURE\_CREDENTIALS }}

- name: Set up Azure Functions Core Tools

uses: azure/functions-action@v1

with:

app-name: <FUNTION\_APP\_NAME>

package: <PACKAGE/PATH>

publish-profile: ${{ secrets.AZURE\_FUNCTIONAPP\_PUBLISH\_PROFILE }}

**GCP APP FUNCTION CD TEMPLATE**

name: app function deployment

on:

push:

branches:

[ main ]

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- name: Download artifact from build job

uses: actions/download-artifact@v3

with:

name: <ARTIFACT\_NAME>

path: <PATH\_TO\_DOWNLOAD>

- name: 'Set up Cloud SDK'

uses: google-github-actions/setup-gcloud@v1

with:

version: '>= 363.0.0'

- name: 'Authenticate to Google Cloud'

uses: 'google-github-actions/auth@v1'

with:

credentials\_json: ${{ secrets.GOOGLE\_APPLICATION\_CREDENTIALS }}

- name: print the credentials

run: echo "${{ secrets.GOOGLE\_APPLICATION\_CREDENTIALS }}"

- name: Deploy to Google App Engine

run: |

gcloud app deploy

- name: get-the-app-engine-link

run: |

gcloud app browse -s <service\_name>

**AZURE KUBEREBETES CD TEMPLATE**

name: Azure Kubernetes deployment

on:

push:

branches:

[ main ]

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- name: Download artifact from build job

uses: actions/download-artifact@v3

with:

name: <ARTIFACT\_NAME>

path: <PATH\_TO\_DOWNLOAD>

- name: Azure Login

uses: azure/login@v1

with:

creds: ${{ secrets.AZURE\_CREDENTIALS }}

- name: Fetch Kubeconfig

run: |

az aks get-credentials --resource-group <resource\_group\_name> --name <aks\_cluster\_name>

mkdir -p ${{ github.workspace }}/.kube

cp ~/.kube/config ${{ github.workspace }}/.kube/config

- name: Set up Kubernetes CLI

uses: azure/setup-kubectl@v1

with:

kubeconfig: ${{ githu.workspace }}/.kube

- name: Apply Kubernetes configuration

run: |

kubectl apply -f ${{ github.workspace }}/<path\_to\_kubernetes\_yaml>