Convert the below json to yaml

{

"id": 101,

"firstName": "John",

"lastName": "Doe",

"email": "john.doe@example.com",

"phoneNumbers": [

{"work": "+1234567890"},

{"home": "+0987654321"},

{"alternative": "+1122334455"}

],

"salary": 75000,

"isActive": true,

"address": {

"street": "123 Main St",

"city": "Anytown",

"state": "CA",

"zipCode": "90210",

"country": "USA"

},

"department": {

"id": 10,

"name": "IT",

"location": "Building A”

}

}

Git Playground url: <http://10.32.2.71:8080?gitplayground>

Note: only works in the VM’s provided as lab in the training

# **Git Concepts and Commands Guide**

## **1. Commit**

### **Concept**

A commit in Git records changes to the repository. Each commit has a unique identifier and includes a commit message describing the changes.

### **Step-by-Step**

1. Create a commit:

git commit

* + This command will open your default text editor for you to enter a commit message. Save and close the editor to complete the commit.

## **2. Branches**

### **Concept**

Branches allow you to work on different versions of a project simultaneously. Each branch represents an independent line of development.

### **Step-by-Step**

1. Create a new branch:

git branch dev

* + This creates a new branch named dev.

1. Switch to the new branch:

git checkout dev

* + This command switches to the dev branch.

## **3. Checkout**

### **Concept**

The checkout command is used to switch between branches or to restore files in the working directory to a specific state.

### **Step-by-Step**

1. Switch to a specific commit (detached HEAD state):

git commit

git checkout C1

* + This checks out commit C1, placing you in a detached HEAD state where you're no longer on a branch.

1. Switch to a branch:

git checkout main

* + This switches to the main branch.

## **4. Merge**

### **Concept**

Merging combines changes from one branch into another. It integrates the history of both branches.

### **Step-by-Step**

1. Merge a branch into another:

git checkout -b dev

git commit

git checkout main

git commit

git merge dev

* + git checkout main switches to the main branch.
  + git merge dev integrates changes from the dev branch into main.

## **5. Rebase**

### **Concept**

Rebasing is an alternative to merging that allows you to apply commits from one branch onto another, creating a linear history.

### **Step-by-Step**

1. Rebase one branch onto another:

git checkout -b dev

git commit

git checkout main

git commit

git checkout dev

git rebase main

git checkout main

git merge dev

git commit

git branch -D dev

* + git checkout dev switches to the dev branch.
  + git rebase main re-applies commits from dev on top of the main branch.

## **6. Cherry-Pick**

### **Concept**

Cherry-picking allows you to apply the changes from a specific commit onto your current branch.

### **Step-by-Step**

1. Cherry-pick specific commits:

git commit

git checkout -b dev

git commit

git checkout main

git commit

git checkout dev

git commit

git commit

git checkout main

git cherry-pick C5

git commit

1. First workflow:

name: 02-01-Simple Workflow

on: push

jobs:

my-first-job:

runs-on: ubuntu-latest

steps:

- run: echo "Hello World"

my-second-job:

runs-on: ubuntu-latest

steps:

- run: echo "Bye World"

1. Labeler.yml in .github folder

# Add 'Documentation' label to any file changes within ‘Docs' folder

Documentation:

- changed-files:

- any-glob-to-any-file:

- Docs/\*

1. Contexts github:

name: 08-01-Contexts - github

on:

workflow\_dispatch:

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Display Event Information

run: |

echo "Event Name: ${{ github.event\_name }}"

echo "Ref: ${{ github.ref }}"

echo "SHA: ${{ github.sha }}"

echo "Actor: ${{ github.actor }}"

echo "Workflow: ${{ github.workflow }}"

echo "Run ID: ${{ github.run\_id }}"

echo "Run number: ${{ github.run\_number }}"

2. Environment context

name: 08-02-Contexts - env

on:

workflow\_dispatch:

env:

MY\_WORKFLOW\_VAR: 'workflow'

MY\_OVERWRITTEN\_VAR: 'workflow'

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Print Env Variables

env:

MY\_OVERWRITTEN\_VAR: 'step'

run: |

echo "Workflow env: ${{ env.MY\_WORKFLOW\_VAR }}"

echo "Overwritten env: ${{ env.MY\_OVERWRITTEN\_VAR }}"

- name: Print Env Variables

run: |

echo "Workflow env: ${{ env.MY\_WORKFLOW\_VAR }}"

echo "Overwritten env: ${{ env.MY\_OVERWRITTEN\_VAR }}"

3. Input context:

name: 08-03-Contexts - Inputs

on:

workflow\_dispatch:

inputs:

somevalue:

type: string

debug:

type: boolean

default: false

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Use Workflow Input

run: |

echo "DEBUG: ${{ inputs.debug }}"

1. Outputs context:

name: 08-04-Contexts - Outputs

on:

workflow\_dispatch:

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Define Workflow Variable

id: set\_var

run: |

echo "Setting Workflow Variable"

echo "name=Max" >> "$GITHUB\_OUTPUT"

- name: Use Workflow Variable

run: |

echo "Workflow Variable Value: ${{ steps.set\_var.outputs.name }}"

Activity Filters:

name: 07-01-Event Filters and Activity Types

on:

pull\_request:

types:

- opened

- synchronize

branches:

- main

jobs:

echo:

runs-on: ubuntu-latest

steps:

- run: echo "Running whenever a PR is opened or synchronized AND base branch is main"

Expressions:

name: 09-01-Using Expressions

run-name: 09-01-Using Expressions | DEBUG - ${{ inputs.debug && 'ON' || 'OFF' }}

on:

push:

workflow\_dispatch:

inputs:

debug:

type: boolean

default: false

jobs:

echo:

runs-on: ubuntu-latest

steps:

- name: '[debug] Print start-up data'

if: inputs.debug

run: |

echo "Triggered by: ${{ github.event\_name }}"

echo "Branch: ${{ github.ref }}"

echo "Commit SHA: ${{ github.sha }}"

echo "Runner OS: ${{ runner.os }}"

- name: '[debug] Print when triggered from master'

if: inputs.debug && github.ref == 'refs/heads/master'

run: echo "I was triggered from master"

- name: Greeting

run: echo "Hello, world"

Functions:

name: 11-01-Using-Functions

on:

pull\_request:

workflow\_dispatch:

jobs:

echo1:

runs-on: ubuntu-latest

steps:

- name: Print PR title

run: echo "${{ github.event.pull\_request.title }}"

- name: Print PR labels

run: |

cat << EOF

${{ toJSON(github.event.pull\_request.labels) }}

EOF

- name: Bug step

if: ${{ !cancelled() && contains(github.event.pull\_request.title, 'fix') }}

run: echo "I am a bug fix"

- name: Sleep for 20 seconds

run: sleep 20

- name: Failing step

run: exit 1

- name: I will be skipped

if: ${{ success() }}

run: echo "I will print if previous steps succeed."

- name: I will execute

if: ${{ failure() }}

run: echo "I will print if any previous step fails."

- name: I will execute

if: ${{ !cancelled() }}

run: echo "I will always print, except when the workflow is cancelled."

- name: I will execute when cancelled

if: ${{ cancelled() }}

run: echo "I will print if the workflow has been cancelled."

Execution Flow:

name: 12-01-Controlling the Execution Flow

on:

workflow\_dispatch:

inputs:

pass-unit-tests:

type: boolean

description: Whether unit tests will pass or not

default: true

jobs:

lint-build:

runs-on: ubuntu-latest

steps:

- name: Lint and build

run: echo "Linting and building project"

unit-tests:

runs-on: ubuntu-latest

steps:

- name: Running unit tests

run: echo "Running tests..."

- name: Failing tests

if: ${{ !inputs.pass-unit-tests }}

run: exit 1

deploy-nonprod:

runs-on: ubuntu-latest

needs:

- lint-build

- unit-tests

steps:

- name: Deploying to nonprod

run: echo "Deploying to nonprod..."

e2e-tests:

runs-on: ubuntu-latest

needs:

- deploy-nonprod

steps:

- name: Running E2E tests

run: echo "Running E2E tests"

load-tests:

runs-on: ubuntu-latest

needs:

- deploy-nonprod

steps:

- name: Running load tests

run: echo "Running load tests"

deploy-prod:

runs-on: ubuntu-latest

needs:

- e2e-tests

- load-tests

steps:

- name: Deploying to prod

run: echo "Deploying to prod..."

**Custom actions:**

Composite Custom Action:

Action.yaml file:

—---------------------------------------------------------------------------------------------------

name: ANZ Setup Java

description: This action allows caching both Java and Maven Dependencies based on the pom.xml file

inputs:

java-version:

description: 'Java version to use'

default: '11'

required: true

working-dir:

description: The working directory of the application

default: .

required: false

runs:

using: 'composite'

steps:

# Step 1: Setup Java version

- name: Setup Java version ${{ inputs.java-version }}

uses: actions/setup-java@v4

with:

distribution: 'adopt'

java-version: ${{ inputs.java-version }}

cache: 'maven' # Cache Maven dependencies

# Step 2: Install Dependencies

- name: Install Dependencies

run: mvn dependency:go-offline

shell: bash

working-directory: ${{ inputs.working-dir }} # Set working directory for Maven command

—----------------------------------------------------------------------------------------------------------------------------

Customactions-composite.yml

—----------------------------------------------------------------------------------------------------------------------------

name: CustomActions-Composite

run-name: CustomActions-Composite

on:

workflow\_dispatch:

env:

working-directory: ./ # Default working directory

jobs:

build:

runs-on: ubuntu-latest

steps:

# Step 1: Checkout repository

- name: checkout repository

uses: actions/checkout@v4

# Step 2: Setup Java & Maven dependencies

- name: Setup Java & Maven dependencies

uses: ./.github/actions/composite-cache-deps

with:

java-version: 11

working-dir: ${{ env.working-directory }} # Use specified working directory

# Step 3: Run tests

- name: Run tests

run: mvn test

# Step 4: Build with Maven

- name: Build with Maven

run: mvn clean install

—----------------------------------------------------------------------------------------------------------------------------

Docker Custom Action:

main.py

—----------------------------------------------------------------------------------------------------------------------------

import os

import requests

import time

def ping\_url(url,delay,max\_trials):

trials = 0

while trials < max\_trials:

try:

response = requests.get(url)

if response.status\_code == 200:

print(f"Website {url} is reachable.")

return True

except requests.ConnectionError:

print(f"Website {url} is unreachable. Retrying in {delay} seconds...")

time.sleep(delay)

trials += 1

except requests.exceptions.MissingSchema:

print(f"Invalid URL format: {url}. Make sure the URL has a valid schema (e.g., http:// or https://)")

return False

return False

def run():

website\_url = os.getenv("INPUT\_URL")

delay = int(os.getenv("INPUT\_DELAY"))

max\_trials = int(os.getenv("INPUT\_MAX\_TRIALS"))

website\_reachable = ping\_url(website\_url,delay,max\_trials)

if not website\_reachable:

raise Exception(f"Website {website\_url} is malformed or unreachable.")

print(f"Website {website\_url} is reachable.")

if \_\_name\_\_ == "\_\_main\_\_":

run()

—----------------------------------------------------------------------------------------------------------------------------

Dockerfile

—----------------------------------------------------------------------------------------------------------------------------

FROM python:alpine3.19

WORKDIR /app

COPY requirements.txt .

RUN python -V

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD [ "python", "/app/main.py" ]

—----------------------------------------------------------------------------------------------------------------------------

requirements.txt

—----------------------------------------------------------------------------------------------------------------------------

certifi==2023.11.17

charset-normalizer==3.3.2

idna==3.6

requests==2.31.0

urllib3==2.1.0

—----------------------------------------------------------------------------------------------------------------------------

action.yml

—----------------------------------------------------------------------------------------------------------------------------

name: Ping URL

description: Ping URL until maximum trials have exceeded. If result is not 200 until then, fails the action.

inputs:

url: # INPUT\_URL

description: URL to ping.

required: true

max\_trials: # INPUT\_MAX-TRIALS

description: Maximum number of trials until action fails.

default: '10'

required: false

delay:

description: Delay in seconds between trials.

default: '5'

required: false

runs:

using: docker

image: Dockerfile

—----------------------------------------------------------------------------------------------------------------------------

CustomActions-Docker.yaml

—----------------------------------------------------------------------------------------------------------------------------

name: CustomActions-Docker

on:

workflow\_dispatch:

inputs:

url:

type: choice

options:

- 'https://www.google.com'

- 'https://www.invalid-aidmdkjmcdkx.com'

- 'invalid-aidmdkjmcdkx'

max\_trials:

description: Maximum trials until action fails.

default: '10'

required: false

delay:

description: Delay in seconds between trials.

default: '5'

required: false

jobs:

ping-url:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- name: Ping URL

uses: ./.github/actions/docker-ping-url

with:

url: ${{ inputs.url }}

max\_trials: ${{ inputs.max\_trials }}

delay: ${{ inputs.delay }}

—----------------------------------------------------------------------------------------------------------------------------

Self Hosted Runners

Run the below command on powershell after opening as run as administrator:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned

Selfhosted.yml

—----------------------------------------------------------------------------------------------------------------------------

name: selfhostedrunner

on: workflow\_dispatch

jobs:

my-first-job:

runs-on: self-hosted

steps:

- run: dir

—----------------------------------------------------------------------------------------------------------------------------

Cacheexample.yml

—----------------------------------------------------------------------------------------------------------------------------

name: 15-01-Caching

on:

workflow\_dispatch

jobs:

install-dependencies:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Java

uses: actions/setup-java@v4

with:

distribution: 'adopt'

java-version: '11'

cache: 'maven'

- name: Install Dependencies

run: mvn dependency:go-offline

test:

runs-on: ubuntu-latest

needs: install-dependencies

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Java

uses: actions/setup-java@v4

with:

distribution: 'adopt'

java-version: '11'

cache: 'maven'

- name: Test with Maven

run: mvn -B test

build:

runs-on: ubuntu-latest

needs: install-dependencies

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Java

uses: actions/setup-java@v4

with:

distribution: 'adopt'

java-version: '11'

cache: 'maven'

- name: Build with Maven

run: mvn clean install -Dmaven.test.skip=true

—----------------------------------------------------------------------------------------------------------------------------

Artifacts.yml

—----------------------------------------------------------------------------------------------------------------------------

name: Artifacts

on:

workflow\_dispatch

jobs:

upload-artifact:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Java

uses: actions/setup-java@v4

with:

distribution: 'adopt'

java-version: '11'

cache: 'maven'

- name: Build with Maven

run: mvn clean install

- run: mkdir staging && cp target/\*.jar staging

- uses: actions/upload-artifact@v4

with:

name: Package

path: staging

download-artifact:

runs-on: ubuntu-latest

needs: upload-artifact

steps:

- name: Download web-app content

uses: actions/download-artifact@v4

with:

name: Package

- name: View content

run: ls -R

—----------------------------------------------------------------------------------------------------------------------------

Releases.yml

—----------------------------------------------------------------------------------------------------------------------------

name: Release

on:

workflow\_dispatch

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Java

uses: actions/setup-java@v4

with:

distribution: 'adopt'

java-version: '11'

cache: 'maven'

- name: Build with Maven

run: mvn clean install -Dmaven.test.skip=true

- run: mkdir staging && cp target/\*.jar staging

- uses: actions/upload-artifact@v4

with:

name: Package

path: staging

release:

runs-on: ubuntu-latest

needs: build

steps:

- name: Download web-app content

uses: actions/download-artifact@v4

with:

name: Package

- name: View content

run: ls -R

- name: Archive site content

uses: thedoctor0/zip-release@master

with:

filename: app.zip

- name: Create GitHub release

id: create-new-release

uses: actions/create-release@v1

env:

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

with:

tag\_name: ${{ github.run\_number }}

release\_name: Release ${{ github.run\_number }}

- name: Upload release asset

uses: actions/upload-release-asset@v1

env:

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

with:

upload\_url: ${{ steps.create-new-release.outputs.upload\_url }}

asset\_path: ./app.zip

asset\_name: app-v${{ github.ref\_type }}.zip

asset\_content\_type: application/zip

—----------------------------------------------------------------------------------------------------------------------------

Matrices.yml

—----------------------------------------------------------------------------------------------------------------------------

# Workflow for Testing applications on different environments in a single workflow

name: Matrices

on:

workflow\_dispatch

jobs:

backwards-compatibility:

name: ${{ matrix.os }}-${{ matrix.java-version }}

runs-on: ${{ matrix.os }}

strategy:

matrix:

java-version: [11,14,17]

os:

- ubuntu-latest

- windows-latest

steps:

- name: Setup Java

uses: actions/setup-java@v4

with:

distribution: 'adopt'

java-version: ${{ matrix.java-version }}

- name: Perform some tests

run: echo "Running tests on OS ${{ matrix.os }} and Java ${{ matrix.java-version }}"

—----------------------------------------------------------------------------------------------------------------------------

shellscript.yml

—----------------------------------------------------------------------------------------------------------------------------

name: Shell-Script

on:

workflow\_dispatch

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Make script executable and run

run: |

chmod +x ./init.sh

./init.sh

shell: bash

—----------------------------------------------------------------------------------------------------------------------------

APICall.yml

—----------------------------------------------------------------------------------------------------------------------------

name: Fetch-Data-API

on:

workflow\_dispatch:

jobs:

fetch-todo:

runs-on: ubuntu-latest

steps:

- name: Fetch Todo Data

run: |

# Use curl to fetch the data from the API endpoint

# Integrate external systems within the Actions workflow

# prepare API request with the required inputs like token payload etc

todo\_data=$(curl -s https://jsonplaceholder.typicode.com/todos/1)

# Print the fetched data

echo "Fetched Todo Data: $todo\_data"

—----------------------------------------------------------------------------------------------------------------------------

docker.yml

—----------------------------------------------------------------------------------------------------------------------------

name: Docker

on:

workflow\_dispatch:

jobs:

docker-build:

runs-on: ubuntu-latest

steps:

- name: Checkout repository

uses: actions/checkout@v2

- name: Set up JDK

uses: actions/setup-java@v2

with:

java-version: '11'

distribution: 'adopt'

cache: maven

- name: Build with Maven

run: mvn clean install

- name: Build Docker image

run: docker build -t username/app .

- name: Log in to Docker Hub

run: echo "${{ secrets.DOCKER\_PASSWORD }}" | docker login -u "${{ secrets.DOCKER\_USERNAME }}" --password-stdin

- name: Push image to Docker Hub

run: docker push username/app

—----------------------------------------------------------------------------------------------------------------------------

In Command Prompt: docker run -d –rm -p 9090:8081 username:imagename:version

In browser: localhost:9090\swagger-ui\index.html

In Command Prompt:

$ notepad kubeyml.yml

$ kubectl apply -f. \kubeyml.yml

$ kubectl get pods

$ kubectl delete -f. \kubeyml.yml

—----------------------------------------------------------------------------------------------------------------------------

CI Pipeline.yml

—----------------------------------------------------------------------------------------------------------------------------

—----------------------------------------------------------------------------------------------------------------------------

CD Pipeline.yml

—----------------------------------------------------------------------------------------------------------------------------

—----------------------------------------------------------------------------------------------------------------------------