## Jobs

## .github/workflows/My_GitHub.yml

```
# This is a basic workflow to help you get started with Actions

name: CI

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the "main"
branch
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in
parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can
access it
      - uses: actions/checkout@v2

      # Runs a single command using the runners shell
      - name: Run a one-line script
        run: echo Hello, world!

      # Runs a set of commands using the runners shell
      - name: Run a multi-line script
        run: |
          echo Add other actions to build,
          echo test, and deploy your project.
```

```
  existing_job:
    runs-on: ubuntu-latest
    steps:
      - name: Check out code
        uses: actions/checkout@v2

      - name: Run existing command
        run: echo "This is an existing job."

  run-bash-script:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Run Bash Script
        run: |
          #!/bin/bash

          # Получаем путь к директории, где находится скрипт
          BASE_DIR="$(cd "$(dirname "$0")/.." && pwd)"

          # Определяем директории
          BASH_COMAND_DIR="$BASE_DIR/bash_comand"
          BUG_REPORT_DIR="$BASE_DIR/bug_report"
          CLIENT_SERVER_DIR="$BASE_DIR/client_server"
          SQL_DIR="$BASE_DIR/sql"
          CHARLES_DIR="$BASE_DIR/charles"
          POSTMAN_DIR="$BASE_DIR/postman"
          TEST_CASES_DIR="$BASE_DIR/test_cases"

          # Проверка существования папок
          for dir in "$BASH_COMAND_DIR" "$BUG_REPORT_DIR"
"$CLIENT_SERVER_DIR" "$SQL_DIR" "$CHARLES_DIR" "$POSTMAN_DIR"
"$TEST_CASES_DIR"; do
            if [ -d "$dir" ]; then
                echo "Папка $(basename "$dir") найдена."
            else
                echo "Папка $(basename "$dir") не найдена."
            fi
          done
```

## Описание:

В YAML-файле описан рабочий процесс CI (непрерывной интеграции), который включает в себя три джоба.

1. Джоб: **build**
Описание: Этот джоб отвечает за начальную сборку проекта. Он выполняется на последней версии Ubuntu.
Используется действие actions/checkout@v2 для получения кода из репозитория, что позволяет дальнейшие шаги взаимодействовать с исходным кодом.
Однострочный скрипт: Выполняется команда, которая просто выводит текст "Hello, world!".
Многострочный скрипт: Выполняется набор команд, который выводит несколько строк текста, показывая, что можно добавлять другие действия для сборки, тестирования и развертывания проекта.

2. Джоб: **existing_job**
Описание: Этот джоб также выполняется на Ubuntu и служит для демонстрации существующей команды.
Как и в предыдущем джобе, используется действие actions/checkout@v2 для доступа к коду репозитория.
Существующая команда: Выполняется команда, которая выводит текст "This is an existing job." Это может быть полезно для проверки простых команд или демонстрации существующей функциональности.

3. Джоб: **run-bash-script**
Описание: Этот джоб предназначен для выполнения Bash-скрипта, который проверяет наличие определенных директорий в проекте.
Шаги:
Checkout: Снова осуществляется проверка кода через действие actions/checkout@v2.
Запуск Bash-скрипта:
Скрипт получает путь к директории, где он находится.
Он определяет несколько директорий, в которых нужно проверить существование.
Затем выполняется цикл, который обходит все указанные директории и выводит сообщение о том, найдены ли они или нет.

**Итог:**
В рабочем процессе CI реализованы три джоба: build, existing_job и run-bash-script. Каждый из них выполняет разные функции:

**build:** Основной джоб, который демонстрирует базовые операции сборки.
**existing_job:** Простой джоб для демонстрации существующих команд.
**run-bash-script:** Более сложный джоб, проверяющий структуру директорий.
Эти джобы могут работать как последовательно, так и параллельно, в зависимости от потребностей и конфигурации рабочего процесса.