



Piotr Rogala



Principal Architect, Azure





MVP Azure



Group leader





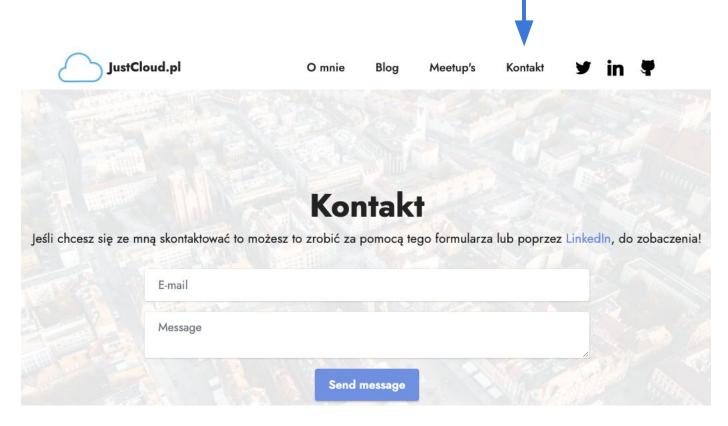


@RogalaPiotr



linkedin.com/in/rogalapiotr

Questions?



Agenda

- 1. Wstęp do ADO i GH
- 2. Przykładowy projekt
- 3. Opcje "to do"
 - a. ChatGPT
 - b. GH Importer
 - c. inne
- 4. Podsumowanie





Wstęp do ADO i GH
Co jest co



GitHub i Azure DevOps

Comparative	GitHub	Azure Devops
Boards	Backlog and Kanban boards with GitHub Issues and GitHub Project	More intuitive board for Scrum development with Azure Boards.
Repository management	GitHub Repositories	Azure Repos
Integrated development	Codespaces	Visual Stidio Code / Visual Studio
Implementation model	Cloud u OnPremise	Cloud u OnPremise
Active Directory integration	Enterprise model only	Native integration
Testing	N/A	Azure Test Plans
CI/CD	GitHub Actions	Azure Pipelines
Documentation	GitHub Wiki	Azure Wiki
Permission management	High level for collaborators (reading/writing)	Granular management of permissions to collaborators of various levels
Reports	GitHub Insights	Azure reporting / Dashboards

GitHub i Azure DevOps

- Repozytorium
- Proces CI/CD
- Integracja / wsparcie dla Azure oraz innymi vendorami
- Agenci wdrożeniowi Agents
- Standalone Server option
- Enterprise Azure Entra

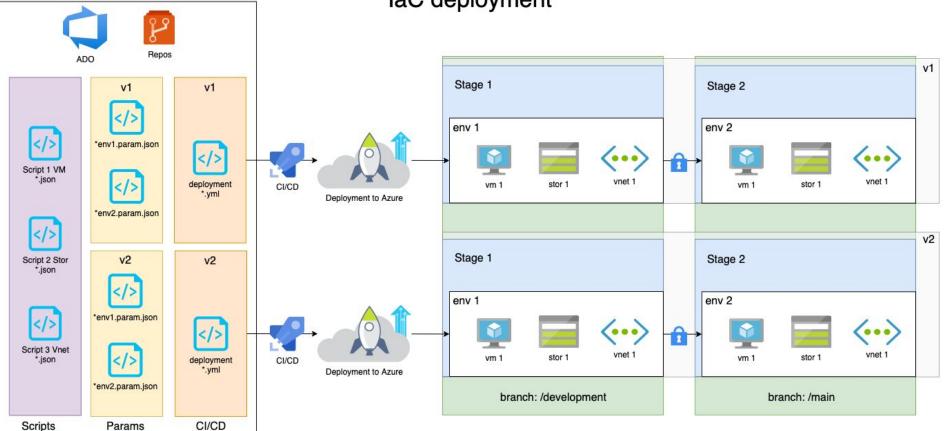
Przykładowy projekt



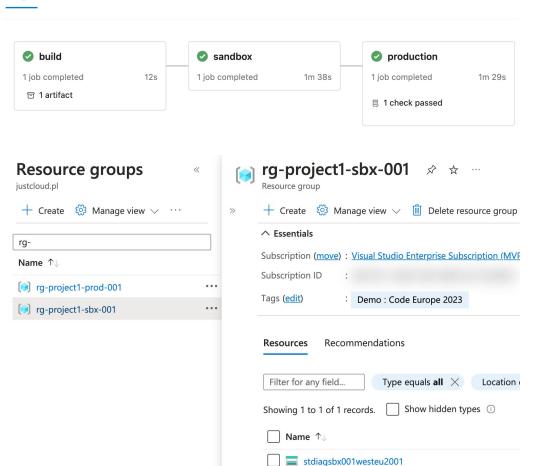
Projekt - ważne kwestie

- Obecnie wdrożone rozwiązanie w Azure DevOps
- Przykład wdrożenie szablonu ARM
- YAML zawiera:
 - Główny plik ze stages sandbox + production
 - Build
 - Deployment
- Użyć tę samą strategię w GitHub
- Szybkie przeniesienie wdrożenie do GitHub w tej samej postaci

IaC deployment



Stages Jobs





Opcje "to do" ChatGPT, Importer, inne?

Scenariusz

- Mamy istniejący projekt w ADO
- Chcemy łatwo przenieść istniejący proces z ADO do GH
- Sprawdzamy ChatGPT
- Sprawdzamy GitHub Importer
- Sprawdzamy inne opcje?



ChatGPT

- 1. Przygotowanie promptu
- 2. Zweryfikowanie odpowiedzi
- 3. Test?





Oczywiście, możemy dostosować Twoje pliki YAML w celu wykorzystania GitHub Actions do wdrażania szablonów ARM do Azure. Oto ekwiwalentne pliki YAML dla GitHub Actions:

Plik project-1.yml

- name: Build

- name: Publish Artifact

run: echo "Perform the build here"

uses: actions/upload-artifact@v2

```
yaml

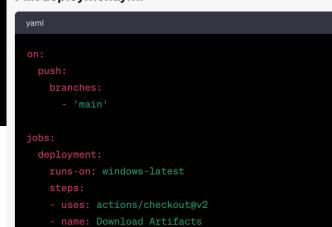
copy code

name: ${{ github.event_name }}

on:
    workflow_dispatch:
    push:
    branches:
    - 'main'

jobs:
    build:
    runs-on: windows-latest
    steps:
    - uses: actions/checkout@v2
Plik deployment.yml
```

Prompt: W Azure DevOps Pipelines wykorzystuje do wdrożenia szablonów ARM do Azure. Na podstawie plików YAML chciałbym zebys zaproponował wdrożenie tych samych szablonów do Azure z GitHub Action. Poniższe skrypty działają w następujący sposób: project-1.yml jest plikiem nadrzednym, plik build.yml jest plikiem podrzednym oraz plik deployment.yml jest plikiem podrzędnym. Chciałbym zachowac te sama strategie wdrozenia.



GitHub Importer

- 1. Automat do migracji procesów
 - a. Wsparcie ADO i wiele innych
- 2. Dokumentacja







Importing 2534 commits...

89% complete

```
name: warsztaty/project-1-demo
                                                               Command: gh actions-importer migrate
 workflow_dispatch:
env:
                                                               azure-devops pipeline --pipeline-id 73
 ProductionSubscriptionID: 3e8197c1-ba00-4526-9bf8-e07273aefbc7
 SandboxSubscriptionID: 3e8197c1-ba00-4526-9bf8-e07273aefbc7
                                                               --target-url
 agentImage: windows-latest
 agentPool: Azure Pipelines
                                                               https://github.com/RogalaPiotrWorkshops/
 artifacts: project-1
 location: westeurope
                                                              github-migration --output-dir tmp/migrate
 serviceConnectionProd: Sandbox
 serviceConnectionSandbox: Sandbox
 project_1_ci_build:
   name: project_1_ci_build
                                                    name: project_1_ci_deployment
   uses: "./.github/workflows/project 1 ci build.yml"
                                                      workflow call:
     agentPool: "${{ env.agentPool }}"
     artifacts: "${{ env.artifacts }}"
                                                         serviceConnection:
 project_1_ci_deployment:
                                                          agentPool: --
   name: project_1_ci_deployment
                                                         agentImage: -
   needs: project 1 ci build
                                                         artifacts:
   uses: "./.github/workflows/project_1_ci_deployment.yml
                                                          deploymentStage: -
                                                          subscriptionID: -
                                                         location: -
                                                     iobs:
                                                      deployment_${{ parameters.deploymentStage }}:
                                                        name: Deployment on env ${{ parameters.deploymentStage }}
                                                        runs-on: ${{ inputs.agentImage }}
                                                        environment:
                                                          name: azure-approval-${{ inputs.deploymentStage }}
                                                        if: github.RUN NUMBER == 1
```

Czy to jest proste przejście z ADO do GH?



Przepisanie procesu od nowa

- Zweryfikowanie możliwości ADO vs GH
- Wykorzystanie funkcjonalności GH
- Zachowanie procesu deploymentu, ale wymagana zmiana kodu

```
name: deployment-2-main-file
iobs:
 release_sandbox:
     DEPLOYMENTSTAGE: sandbox
     SUBSCRIPTIONID: ${{ vars.SANDBOXSUBSCRIPTIONID }}
 release_production:
   uses: ./.github/workflows/deployment-2-file-second.yml
     DEPLOYMENTSTAGE: production
     SUBSCRIPTIONID: ${{ vars.SANDBOXSUBSCRIPTIONID }}
```

```
DEPLOYMENTSTAGE:
      required: true
      type: string
    SUBSCRIPTIONID:
      required: true
      type: string
deploy:
    name: Checkout repository
      uses: actions/checkout@v2
    - name: Set up Azure CLI
      uses: azure/login@v1
      with:
        creds: ${{ secrets.AZURE_CREDENTIALS }}
    # Deploy resource group
    - name: deploy-resource-group
      id: deploy-1
```

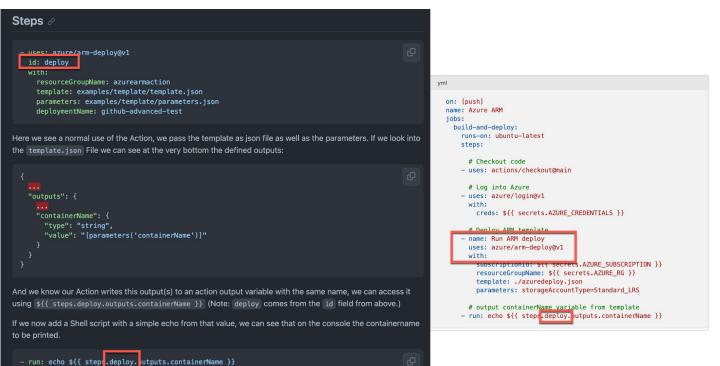
Błędy które napotkałem

- 1. Params / Env
 - a. Vars jako security
 - b. Przekazywanie ENV między szablonami
- 2. Build / Checkout
- 3. Dokumentacja (ChatGPT bazuje na tym)
 - a. MS nie wspomina o wymaganym ID: https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-github-actions?tabs=userlevel
 - b. GH tak https://github.com/Azure/arm-deploy#steps-1
- 4. Przechwytywanie Outputs z szablonów



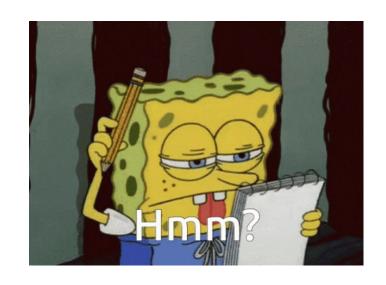
1. Dokumentacja (ChatGPT bazuje na tym) - missing part

- a. MS nie wspomina o wymaganym ID: https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-github-actions?tabs=userlevel
- b. GH tak <u>https://github.com/Azure/arm-deploy#steps-1</u>



Podsumowanie

- ChatGPT pomocny, ale tylko pomocny
- GitHub Importer fajny, ale...
- Warto sprawdzić przed przepisywaniem
- Im bardziej założone taski (custom code) lub zagnieżdżenia będą trudniejsze do przeniesienia
- Wymagana zmiana podejścia do
 - o Gates / PR
 - Build / Checkout
 - Params / Variables



Questions?

