



Piotr Rogala

Z Azure DevOps do GitHub Actions
z użyciem ChatGPT



Expert Summit 2023

All about Microsoft® technologies

ONLINE

Sala A

Godz. 16:00




Demant



Z Azure DevOps do GitHub Actions z użyciem ChatGPT



 @RogalaPiotr

 justcloud

 Nordcloud
an IBM Company

Piotr Rogala

Working in  **Nordcloud**
an IBM Company

Principal Architect, Azure

Blog  justcloud 

MVP Azure 

Group leader  Microsoft Azure
User Group Poland



@RogalaPiotr



[linkedin.com/in/rogalapiotr](https://www.linkedin.com/in/rogalapiotr)

Questions?



JustCloud.pl

O mnie

Blog

Meetup's

Kontakt



Kontakt

Jeśli chcesz się ze mną skontaktować to możesz to zrobić za pomocą tego formularza lub poprzez [LinkedIn](#), do zobaczenia!

Send message

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



Azure DevOps

GitHub i Azure DevOps

Comparative	CitHub	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

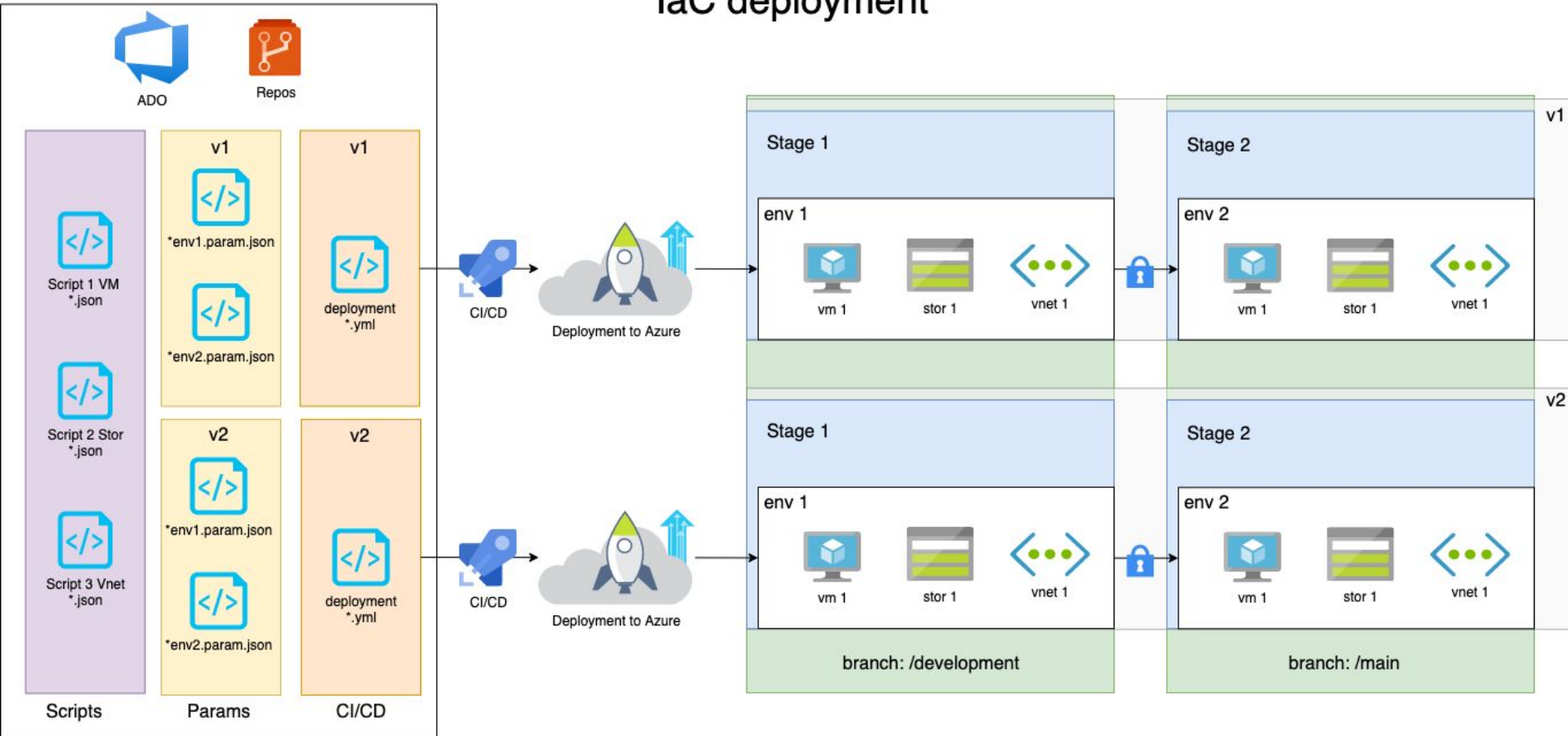
Przykładowy projekt



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 wdrożenia do GitHub w tej samej postaci

IaC deployment



▼ project-1

▼ ci

build.yml

deployment.yml

project-1.yml

▼ scripts

createresourcegroup-project1.production.param.j...

createresourcegroup-project1.sandbox.param.json

createresourcegroup.json

storage-diag001.production.param.json

storage-diag001.sandbox.param.json

storage.json

Stages Jobs

✓ build

1 job completed

12s

1 artifact

✓ sandbox

1 job completed

1m 38s

✓ production

1 job completed

1m 29s

1 check passed

Resource groups

justcloud.pl

+ Create ⚙️ Manage view ▼ ⋮

rg-

Name ↑↓

rg-project1-prod-001

rg-project1-sbx-001



rg-project1-sbx-001

Resource group

+ Create ⚙️ Manage view ▼ 🗑️ Delete resource group

Essentials

Subscription (move) : [Visual Studio Enterprise Subscription \(MVE](#)

Subscription ID :

Tags (edit) : Demo : Code Europe 2023

Resources

Recommendations

Filter for any field...

Type equals all ✕

Location

Showing 1 to 1 of 1 records. ☐ Show hidden types ⓘ

☐ Name ↑↓

☐ stdiagsbx001westeu2001



Opcje “to do”

ChatGPT, Importer, inne?

Scenariusz

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



ChatGPT

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



CHATGPT

<https://chat.openai.com/>



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

yaml

Copy code

```
name: ${ github.event_name }}

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

jobs:
  build:
    runs-on: windows-latest
    steps:
      - uses: actions/checkout@v2
      - name: Build
        run: echo "Perform the build here"
      - name: Publish Artifact
        uses: actions/upload-artifact@v2
```

Prompt: W Azure DevOps Pipelines wykorzystuje do wdrożenia szablonów ARM do Azure. Na podstawie plików YAML chciałbym żebyś zaproponował wdrożenie tych samych szablonów do Azure z GitHub Action. Poniższe skrypty działają w następujący sposób: project-1.yml jest plikiem nadrzędnym, plik build.yml jest plikiem podrzędnym oraz plik deployment.yml jest plikiem podrzędnym. Chciałbym zachować tę samą strategię wdrożenia.

Plik deployment.yml

yaml

```
on:
  push:
    branches:
      - 'main'

jobs:
  deployment:
    runs-on: windows-latest
    steps:
      - uses: actions/checkout@v2
      - name: Download Artifacts
```

GitHub Importer

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

GitHub Importer



Importing 2534 commits...
89% complete



<https://docs.github.com/en/actions/migrating-to-github-actions/automated-migrations/migrating-from-azure-devops-with-github-actions-importer>

```

1 name: warsztaty/project-1-demo
2 on:
3   workflow_dispatch:
4   env:
5     ProductionSubscriptionID: 3e8197c1-ba00-4526-9bf8-e07273aefbc7
6     SandboxSubscriptionID: 3e8197c1-ba00-4526-9bf8-e07273aefbc7
7     agentImage: windows-latest
8     agentPool: Azure Pipelines
9     artifacts: project-1
10    location: westeurope
11    serviceConnectionProd: Sandbox
12    serviceConnectionSandbox: Sandbox
13 jobs:
14   project_1_ci_build:
15     name: project_1_ci_build
16     uses: "../github/workflows/project_1_ci_build.yml"
17     with:
18       agentPool: "${{ env.agentPool }}"
19       agentImage: "${{ env.agentImage }}"
20       artifacts: "${{ env.artifacts }}"
21   project_1_ci_deployment:
22     name: project_1_ci_deployment
23     needs: project_1_ci_build
24     uses: "../github/workflows/project_1_ci_deployment.yml"

```

Command: `gh actions-importer migrate azure-devops pipeline --pipeline-id 73 --target-url https://github.com/RogalaPiotrWorkshops/expertsummit2023 --output-dir tmp/migrate`

```

name: project_1_ci_deployment
on:
  workflow_call:
    inputs:
      serviceConnection: ...
      agentPool: ...
      agentImage: ...
      artifacts: ...
      deploymentStage: ...
      subscriptionID: ...
      location: ...
jobs:
  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

on:
  push:
    branches:
      - main
    paths:
      - '.github/workflows/deployment-2-main-file.yml'
      - '.github/workflows/deployment-2-file-second.yml'

jobs:
  release_sandbox:
    uses: ../.github/workflows/deployment-2-file-second.yml
    with:
      DEPLOYMENTSTAGE: sandbox
      SUBSCRIPTIONID: ${vars.SANDBOXSUBSCRIPTIONID}
      secrets: inherit

  release_production:
    uses: ../.github/workflows/deployment-2-file-second.yml
    with:
      DEPLOYMENTSTAGE: production
      SUBSCRIPTIONID: ${vars.SANDBOXSUBSCRIPTIONID}
      secrets: inherit
```

main.yml

```
on:
  workflow_call:
    inputs:
      DEPLOYMENTSTAGE:
        required: true
        type: string
      SUBSCRIPTIONID:
        required: true
        type: string

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - 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
```

deployment.yml

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 <https://github.com/Azure/arm-deploy#steps-1>

Steps

```
- uses: azure/arm-deploy@v1
  id: deploy
  with:
    resourceGroupName: azurearmaction
    template: examples/template/template.json
    parameters: examples/template/parameters.json
    deploymentName: github-advanced-test
```

Here we see a normal use of the Action, we pass the template as json file as well as the parameters. If we look into the `template.json` File we can see at the very bottom the defined outputs:

```
{
  "outputs": {
    "containerName": {
      "type": "string",
      "value": "[parameters('containerName')]"
    }
  }
}
```

And we know our Action writes this output(s) to an action output variable with the same name, we can access it using `${{ steps.deploy.outputs.containerName }}` (Note: `deploy` comes from the `id` field from above.)

If we now add a Shell script with a simple echo from that value, we can see that on the console the containername to be printed.

```
- run: echo ${{ steps.deploy.outputs.containerName }}
```

```
yaml
on: [push]
name: Azure ARM
jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      # Checkout code
      - uses: actions/checkout@main

      # Log into Azure
      - uses: azure/login@v1
        with:
          creds: ${{ secrets.AZURE_CREDENTIALS }}

      # Deploy ARM template
      - name: Run ARM deploy
        uses: azure/arm-deploy@v1
        with:
          subscriptionId: ${{ secrets.AZURE_SUBSCRIPTION }}
          resourceGroupName: ${{ secrets.AZURE_RG }}
          template: ./azuredeploy.json
          parameters: storageAccountType=Standard_LRS

      # output containerName variable from template
      - run: echo ${{ steps.deploy.outputs.containerName }}
```

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
- Wymagane zmiana podejścia do
 - Gates / PR
 - Build / Checkout
 - Params / Variables



Questions?



JustCloud.pl

O mnie

Blog

Meetup's

Kontakt



Kontakt

Jeśli chcesz się ze mną skontaktować to możesz to zrobić za pomocą tego formularza lub poprzez [LinkedIn](#), do zobaczenia!

Send message

Q&A

