



Azure Summer Camp

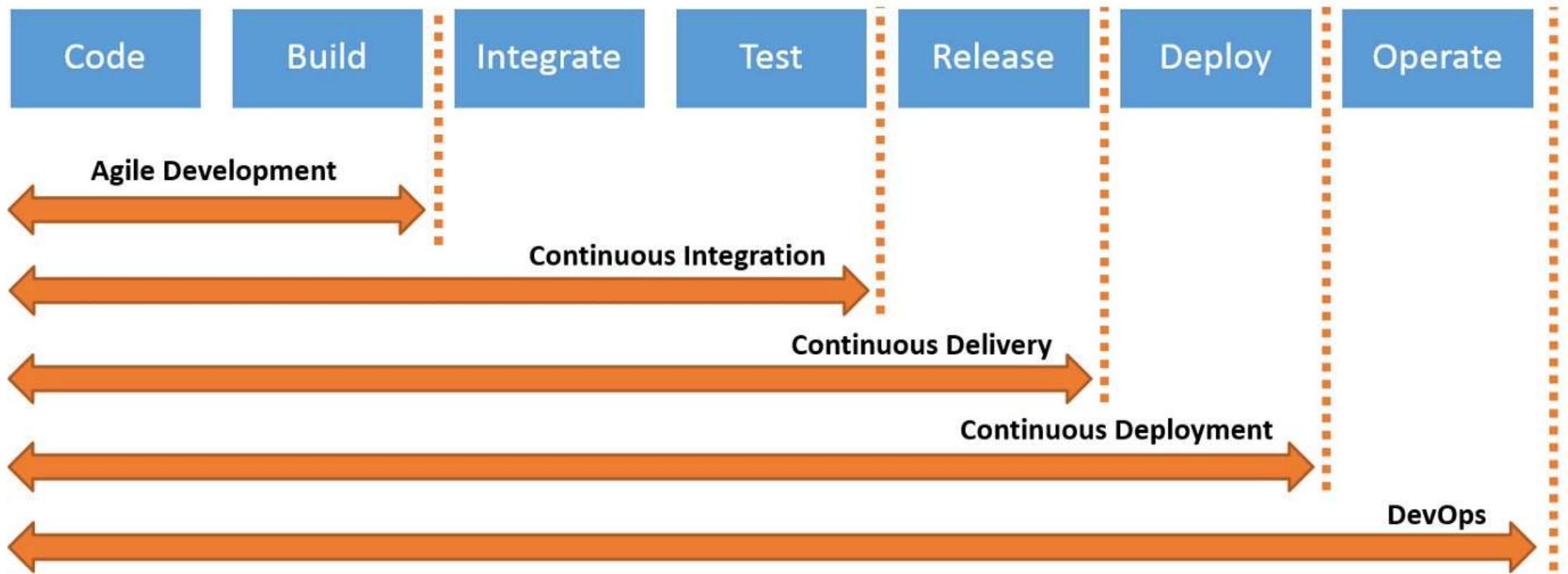
DevOps - Dzień II

Agenda

- 12:00 – 12:25 – Czym jest Azure DevOps i CI/CD.
- 12:25 – 12:50 – Logowanie do projektu i szybki przegląd UI Azure DevOps.
- 12:50 – 13:00 – Przerwa.
- 13:00 – 13:45 - Szybki Release pipeline.
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 1
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 2
- 16:00 – 16:30 – Pytania i zadania dodatkowe

Azure DevOps pozwala na:

- Zarządzanie projektem
- Rozproszoną kontrolę wersji (git)
- Zarządzanie dokumentacją (wiki)
- Continuous Integration
- Continuous Deployment
- Continuous Delivery
- Pełny process wg. Agile i/lub Devops



Agenda

- 12:00 – 12:25 – Czym jest Azure DevOps i CI/CD.
- 12:25 – 12:50 – Logowanie do projektu i szybki przegląd UI Azure DevOps.
- 12:50 – 13:00 – Przerwa.
- 13:00 – 13:45 - Szybki Release pipeline
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 1
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 2
- 16:00 – 16:30 – Pytania i zadania dodatkowe

Wybierz usera, zaloguj się i ustaw hasło

Zakres usernamów: user0001 – user1000

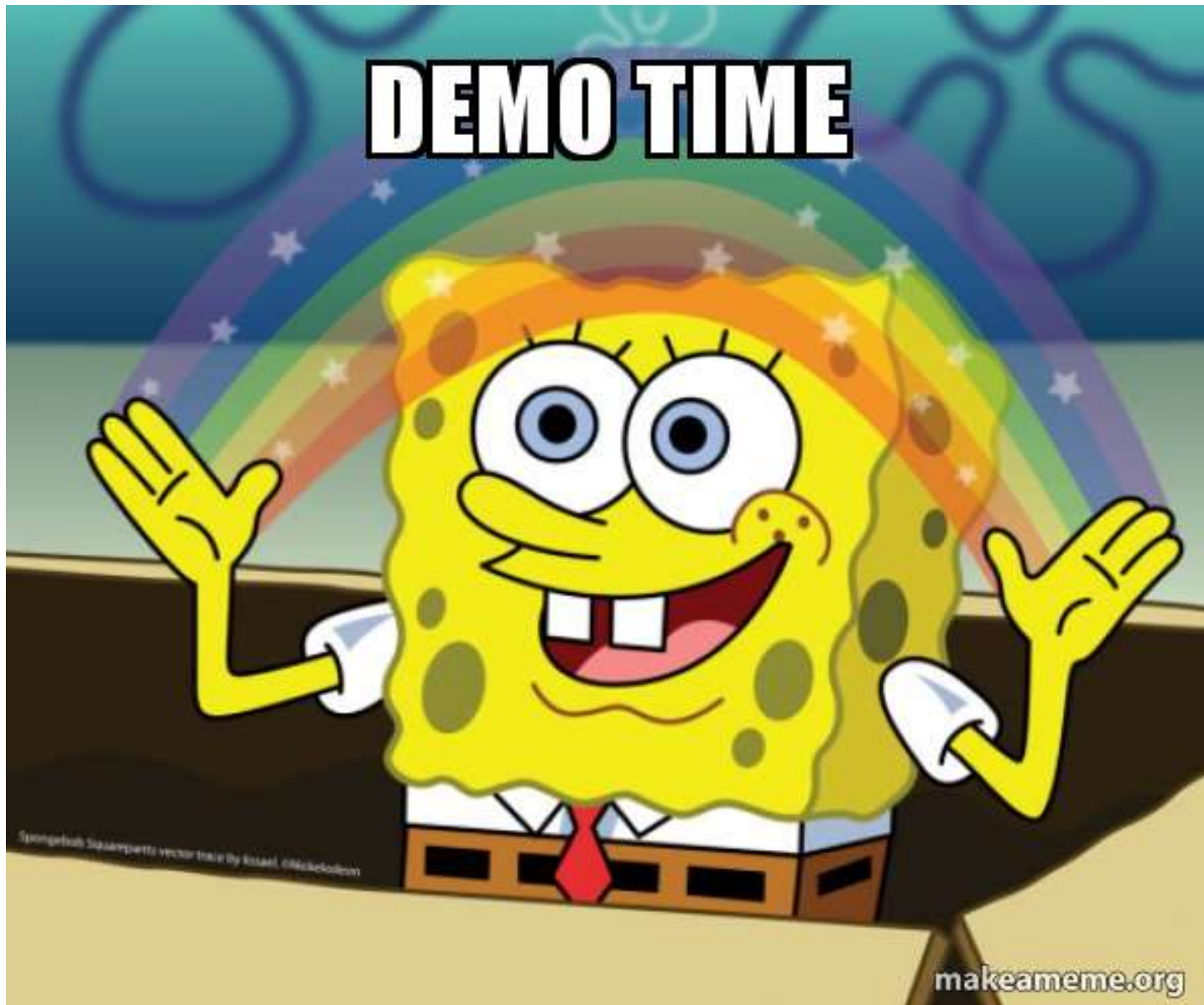
UPN: user12345@rhryniewskibillennium.onmicrosoft.com

Hasło: InitialPass1!2

URL: <https://dev.azure.com/>

Uwaga: Przy pierwszym logowaniu uzupełnij swój mail kontaktowy.

Podczas tworzenia organizacji – jej nazwa = nazwa usera



Zadania

1. Do swojej organizacji dodać userów z uprawnieniami:

Project Administrators

- user1234@rhryniewskibillennium.onmicrosoft.com
- user12345@rhryniewskibillennium.onmicrosoft.com

2. Utworzyć **publiczny** projekt - w sekcji Files zrobić import repozytorium z URLa

<https://github.com/r-hryniewski/BlazorApp>

*. Poklikaj sobie

Agenda

- 12:00 – 12:25 – Czym jest Azure DevOps i CI/CD.
- 12:25 – 12:50 – Logowanie do projektu i szybki przegląd UI Azure DevOps.
- 13:02 – 13:07 – Przerwa.
- 13:00 – 13:45 - Szybki Release pipeline
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 1
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 2
- 16:00 – 16:30 – Pytania i zadania dodatkowe

Dzień 3

Agenda

- ~~• 12:00 – 12:25 – Czym jest Azure DevOps i CI/CD.~~
- ~~• 12:25 – 12:50 – Logowanie do projektu i szybki przegląd UI Azure DevOps.~~
- ~~• 12:50 – 13:00 – Przerwa.~~
- 12:00 – 12:50 - Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 1
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET Cz. 2
- 16:00 – 16:30 – Pytania i zadania dodatkowe

Agenda

- 12:00 – 12:50 - Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET

Wejdź na portal Azure

URL: portal.azure.com

Nazwa RG: Twój username

Nazwa konta storagowego: Twoj username

1. Stwórz resource grupę
2. Stwórz Storage Account
3. Skonfiguruj na nim static file hosting

Na Azure DevOps wejść w releases

Stwórz nowy release pipeline ma on:

- Być triggerowany manualnie
- Artefaktem ma być repozytorium jakie stworzyliśmy
- Ma budować aplikację .net ze ścieżki **/BlazorApp1/BlazorApp1.sln**
- Aplikacja ma być wdrażana na konto storagowe, które stworzyliśmy
Pliki będące artefaktem builda muszą się znaleźć w kontenerze /\$web
- Nadanie uprawnień ->
<https://brettmckenzie.net/2020/03/23/azure-pipelines-copy-files-task-authentication-failed/>

Agenda

- 12:00 – 12:50 – Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET

Agenda

- 12:00 – 12:50 – Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:00 – 13:45 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET

Co to jest yaml?

- Skrót od YAML Ain't Markup Language
- Czytelny i ustrukturyzowany sposób reprezentacji danych
- Powszechny w plikach konfiguracyjnych, CI/CD i w zapisie specyfikacji API

Podstawy składni

- Wpisy zorganizowane są w formie **klucz: wartość**
- Wcięcia (tylko spacje) służą do nadania struktury poprzez zagnieżdżenia.
- Wiersze po myślniku to kolejne element listy
- # - to komentarz

Dobry przewodnik na start:

<https://circleci.com/blog/what-is-yaml-a-beginner-s-guide/>

Schema yamla dla Azure Pipelines:

<https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema/>

Podstawy składni

```
trigger:
- main

pool:
  vmImage: 'windows-latest'

variables:
- solution: '**/*.sln'
  buildPlatform: 'Any CPU'
  buildConfiguration: 'Release'
#komentarz
steps:
- task: NuGetToolInstaller@1
- task: NuGetCommand@2
  inputs:
    restoreSolution: '$(solution)'
- task: VSBUILD@1
  inputs:
    solution: '$(solution)'
    msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true /p:SkipInvalidConfigurations=true /p:PackageLocation=$(BuildDirectory)\PackageLocation'
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'
```

Taski

Dostępne taski:

<https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/>

```
- task: NuGetToolInstaller@1
- task: NuGetCommand@2
  inputs:
    restoreSolution: '$(solution)'
- task: VSBUILD@1
  inputs:
    solution: '$(solution)'
    msbuildArgs: '/p:DeployOnBuild=true /p:'
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'

- task: VSTest@2
  inputs:
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'
```

Zadania

Napisać build pipeline w .yaml (można zacząć od generatora w Azure DevOpsie):

- Ma on budować aplikację .NET tak jak poprzedni, ale nigdzie jej nie wdrażać.
 - Artefakt po buildzie powinien być publikowany
 - Pipeline powinien być uruchamiany po każdej zmianie na branchu main (podpowiedź – trigger)
 - Plik z .yaml wrzucić na repozytorium na branchu main
 - W Azure DevOps stworzyć pipeline i wskazać jego źródło na plik
 - Zmienić cokolwiek na branchu main, by sprawdzić czy wszystko jest ok
- * Użyć pipeline w polityce brancha, by sprawdzać, czy projekt się buduje podczas robienia Pull Requesta

Agenda

- 12:00 – 12:50 – Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:00 – 13:45 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET

Agenda

- 12:00 – 12:50 – Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:00 – 13:45 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 15:00 – 15:10 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET

Zadania (jeśli nie skończyliśmy)

Napisz build pipeline w .yaml:

- Ma on budować aplikację .NET tak jak poprzedni, ale nigdzie jej nie wdrażać.
 - Artefakt po buildzie powinien być publikowany
 - Pipeline powinien być uruchamiany po każdej zmianie na branchu main (podpowiedź – trigger)
 - Plik z .yaml wrzucić na repozytorium na brancha main
 - W Azure DevOps stworzyć pipeline i wskazać jego źródło na plik
 - Zmienić cokolwiek na branchu main, by sprawdzić czy wszystko jest ok
- * Użyj pipeline w polityce brancha, by sprawdzać, czy projekt się buduje podczas robienia Pull Requesta

Agenda

- 12:00 – 12:50 – Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:00 – 13:45 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 15:00 – 15:15 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET

Agenda

- 12:00 – 12:50 – Szybki Release pipeline
- 12:50 – 13:00 – Przerwa.
- 13:00 – 13:45 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 13:45 – 14:15 – Przerwa lunchowa.
- 14:15 – 15:00 – Wstęp do yamla i proste CI/CD aplikacji .NET
- 15:05 – 15:15 – Przerwa
- 15:10 – 16:00 – Wstęp do yamla i proste CI/CD aplikacji .NET

Zadania (jeśli skończyliśmy)

Zmodyfikuj release pipeline

- Powinien być triggerowany po zakończeniu build pipeline napisanego przez Ciebie w .yaml
- Powinien wdrażać artefakt z build pipeline na używany poprzednio storage account

* Przepisz powyższy pipeline na .yaml

SEE YOU LATER

