

Azure Pipelines 補足

- ・パイプラインの種類（クラシックとYAML）
- ・ゲートと承認
- ・変数と式
- ・タスクグループ

パイプラインの種類

クラシック パイプラインとYAMLパイプライン

Azure Pipelinesで作成できるパイプライン

- (旧) クラシック パイプライン

- ビルド パイプライン (ビルド、テストを実行)

+

- リリース パイプライン (リリースを実行)

ビルドパイプラインと
組み合わせ可能
YAMLパイプラインとは
組み合わせ不可

- (新) YAML パイプライン (ビルド、テスト、リリースを実行)

クラシック パイプライン vs YAMLパイプライン

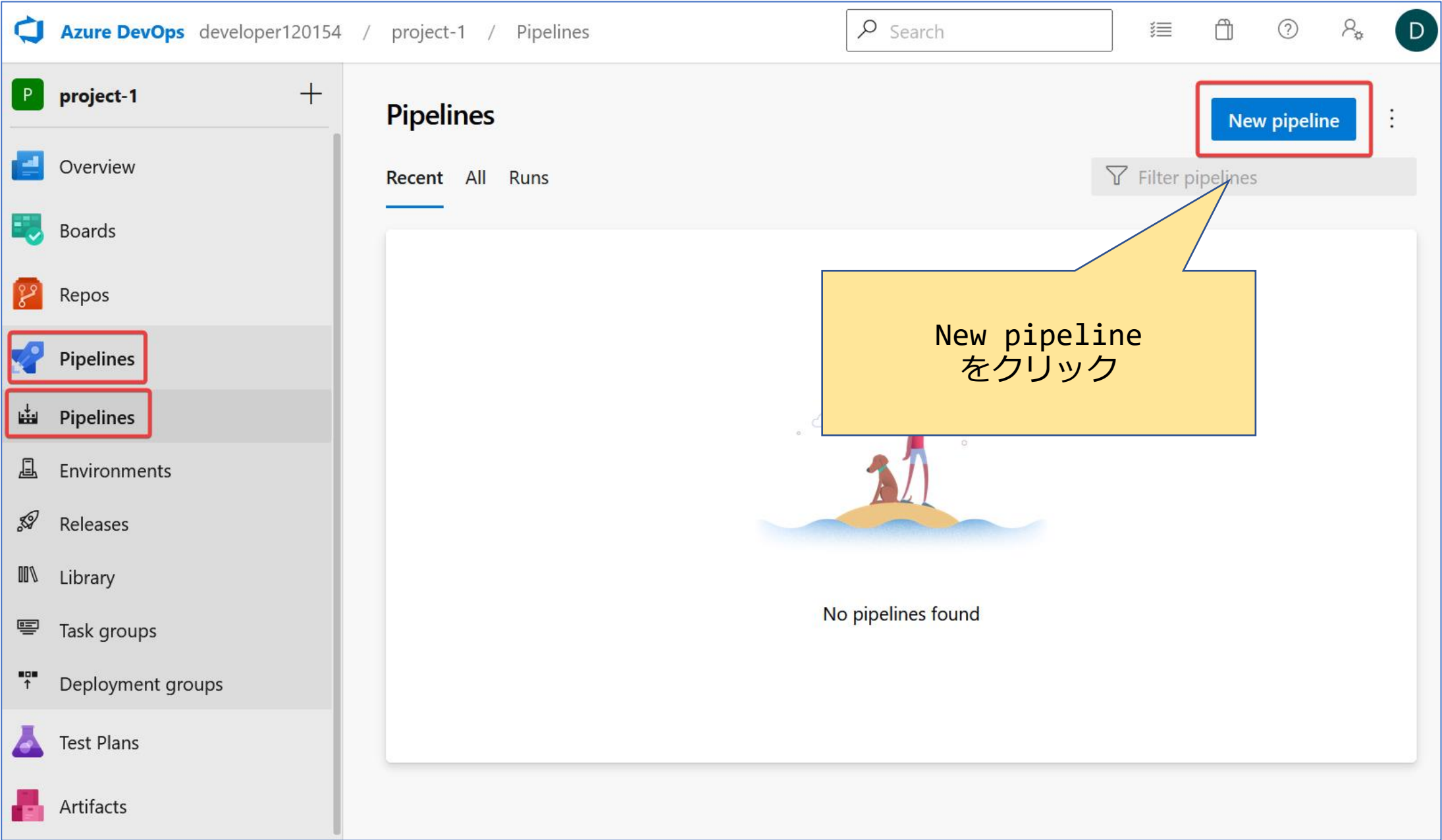
パイプラインの種類	クラシック パイプライン (ビルド パイプライン + リリース パイプライン)	YAMLパイプライン (2017/11/15～)
エディター (ユーザーインターフェース)	グラフィカルエディター	YAMLパイプラインエディター
ビルド・テストと リリースの統一	統一されていない。 ビルドパイプラインと リリースパイプラインで 別のUIを使用。	統一されている。 ビルド・テスト・リリースを 1つのYAMLファイルに記述可。 (2020/4/28～)
CI/CDをコードとして記述・ CI/CDをコードと一緒に保存	できない	できる
パイプライン定義の 保存場所	Azure DevOpsプロジェクト	接続したリポジトリ内の azure-pipelines.yml
移行	YAMLパイプラインへの移行 (YAML エクスポート) が可能	-

<https://devblogs.microsoft.com/devops/pipeline-as-code-yaml-preview/>


<https://devblogs.microsoft.com/devops/announcing-general-availability-of-azure-pipelines-yaml-cd/>

クラシック パイプラインの
ビルド パイプライン






■ 「クラシック パイプライン」の「ビルド パイプライン」の作成




■ 「クラシック パイプライン」の「ビルド パイプライン」の作成


 Azure DevOps


developer120154 / project-1 / Pipelines





 project-1


+


 Overview


 Boards


 Repos


 Pipelines


 Pipelines


 Environments


 Releases


 Library

 Task groups

 Deployment groups

 Test Plans

 Artifacts

 Project settings

<<

Connect


Select

Configure

Review


New pipeline

Where is your code?

 Azure Repos Git


YAML

Free private Git repositories, pull requests, and code search

 Bitbucket Cloud


YAML

Hosted by Atlassian

 GitHub


YAML

Home to the world's largest community of developers


 GitHub Enterprise Server

YAML

The self-hosted version of GitHub Enterprise

 Other Git

Any generic Git repository

 Subversion

Centralized version control by Apache

Use the classic editor

to create a pipeline without YAML.

「クラシックエディターを使用する」
(Use the classic editor)
を選ぶ

■ 「クラシック パイプライン」の「ビルド パイプライン」設定例

project2-ASP.NET Core-CI

Tasks Variables Triggers Options History

Save & queue Discard Summary Queue

Pipeline

Get sources

Agent job 1

Restore

Build

Test

Publish

Publish Artifact

Add tasks

Refresh

Search

All Build Utility Test Package Deploy Tool Marketplace

.NET Core

Android signing

Ant

App Center distribute

App Center test

Name

project2-ASP.NET Core-CI

Agent pool

Azure Pipelines

Agent Specification

ubuntu-latest

Parameters

Unlink all

Project(s) to restore and build

**/*.csproj

Project(s) to test

**/*[!t]ests*

必要事項を記入・選択

クリックでタスクを選択

クラシック パイプラインの
リリース パイプライン

■ 「クラシック パイプライン」の「リリース パイプライン」の作成

Azure DevOps

developer120154 / project4 / Pipelines / Releases

Search

P

project4

+

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Releases
をクリック

No release pipelines found

Automate your release process in a few easy steps with a new pipeline

New pipeline

New pipeline
をクリック

■ 「クラシック パイプライン」の「リリース パイプライン」の設定例

Artifact

Build - _project2-ASP.NET Core-CI

Project * ①

project2

Source (build pipeline) * ①

project2-ASP.NET Core-CI

Default version * ①

Latest

Source alias * ①

_project2-ASP.NET Core-CI

① The artifacts published by each version will be available for the latest successful build of **project2-ASP.NET Core-CI** pipeline **drop**.

Continuous deployment trigger

Build: _project2-ASP.NET Core-CI

☒ Enabled

Creates a release every time a new build is available.

Build branch filters ①

No filters added.

+ Add | v

Pull request trigger

Build: _project2-ASP.NET Core-CI

☐ Disabled

① Enabling this will create a release every time a selected artifact request workflow

Pipeline

Tasks v

Variables

Retention

Options

History

Artifacts | + Add

Stages | + Add v

🏗️

_project2-ASP.NET Core-CI

🕒

Schedule not set

🔗

Stage 1

1 job, 1 task

All pipelines > New release pipeline

Pipeline

Tasks v

Variables

Retention

Options

History

Stage 1

Deployment process

Agent job

Run on agent

🌐

Azure App Service Deploy: bbbb234263456

Azure App Service deploy

Azure App Service deploy ①

View YAML

Remove

Task version 4.* v

Display name *

Azure App Service Deploy: bbbb234263456

Connection type * ①

Azure Resource Manager v

Azure subscription * ① | Manage ↗

App Service type * ①

Web App on Windows v

App Service name * ①

bbbb234263456 v

☐ Deploy to Slot or App Service Environment ①

Virtual application ①

Package or folder * ①

\$(System.DefaultWorkingDirectory)/**/*.zip ...

リリースパイプラインの
ステージ

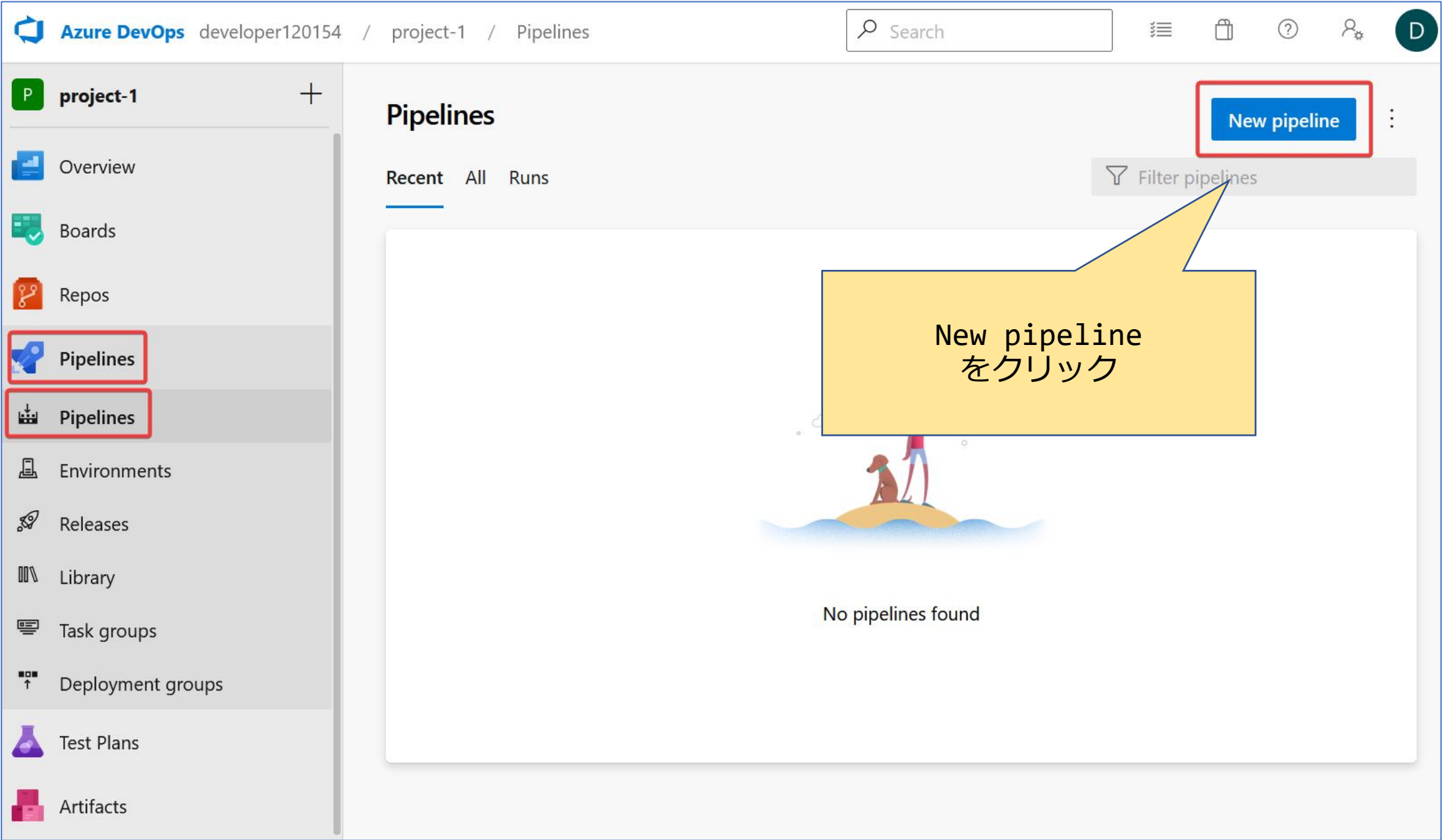
アーティファクト（ビルド
パイプラインの成果物を指
定）

CDトリガー
（ビルドパイプラインが完
了したら
リリースパイプラインを自
動起動）

デプロイタスク

YAMLパイプライン

■ 「YAMLパイプライン」の作成



■ 「YAMLパイプライン」の作成

project4

+

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Azure DevOps

developer120154 / project4 / Pipelines

Search

Connect

Select

Configure

Review

New pipeline

Where is your code?

Azure Repos Git

YAML

Free private Git repositories, pull requests, and code search

Bitbucket Cloud

YAML

Hosted by Atlassian

GitHub

YAML

Home to the world's largest community of developers

GitHub Enterprise Server

YAML

The self-hosted version of GitHub Enterprise

Other Git

Any generic Git repository

Subversion


Centralized version control by Apache

Use the classic editor to create a pipeline without YAML.

これらのどれかを選択すると「YAMLパイプライン」の作成となる

こちらを選ぶと「クラシックパイプライン」の「ビルドパイプライン」作成となる

■ 「YAMLパイプライン」の作成


 Azure DevOps


developer120154 / project5 / Pipelines


P


project5


+

 Overview


 Boards


 Repos


 Pipelines


 Pipelines


Environments


 Releases

 Library

 Task groups

 Deployment groups

 Test Plans

 Artifacts

✓ Connect


Select

Configure

Review


New pipeline

Select a repository

 Filter by keywords


project5

×

 project5

リポジトリを選択

■ 「YAMLパイプライン」の作成

 **Azure DevOps** developer120154 / project5 / Pipelines

P project5 +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans


Artifacts

Search


Connect Select **Configure** Review

New pipeline


Configure your pipeline

 Starter pipeline


Start with a minimal pipeline that you can customize to build and deploy your code.

 Existing Azure Pipelines YAML file


Select an Azure Pipelines YAML file in any branch of the repository.

 .NET Core Function App to Windows on Azure


Build a .NET Core function app and deploy it to Azure as a Windows function App.

 .NET Desktop


Build and run tests for .NET Desktop or Windows classic desktop solutions.

 Android


Build your Android project with Gradle.

 Ant


Build your Java projects and run tests with Apache Ant.

 ASP.NET


Build and test ASP.NET projects.

 ASP.NET Core

Build and test ASP.NET Core projects targeting .NET Core.

 ASP.NET Core (.NET Framework)

Build and test ASP.NET Core projects targeting the full .NET Framework.

 C/C++ with GCC

Build your C/C++ project with GCC using make.

あらかじめ用意されているパイプラインの雛形を選択

■ 「YAMLパイプライン」の作成

✓ Connect

✓ Select

✓ Configure

Review

New pipeline

Review your pipeline YAML

Variables

Save and run

▼

project5 / azure-pipelines.yml *

1 # ASP.NET Core

2 # Build and test ASP.NET Core projects targeting .NET Core.

3 # Add steps that run tests, create a NuGet package, deploy, and more:

4 # <https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core>

5

6 trigger:

7 - main

8

9 pool:

10 | vmImage: ubuntu-latest

11

12 variables:

13 | buildConfiguration: 'Release'

14

15 steps:

16 - script: dotnet build --configuration \$(buildConfiguration)

17 | displayName: 'dotnet build \$(buildConfiguration)'

18

19 - scri

Show assistant

アシスタント

「YAMLパイプラインエディター」
で編集

コード補完、記述ミスの警告などの
入力支援機能あり

script

■ 「YAMLパイプライン」の作成

アシスタントを表示

Variables

and run

Show assistant

Tasks

app ser

App Center distribute

Distribute app builds to testers and users via Visu...

Azure App Service deploy

Deploy to Azure App Service a web, mobile, or AP...

Azure App Service manage

Start, stop, restart, slot swap, slot delete, install sit...

Azure App Service Settings

Update/Add App settings an Azure Web App for ...

Azure Web App for Containers

Deploy containers to Azure App Service

Service Fabric application deployment

Deploy an Azure Service Fabric application to a cl...

Service Fabric Compose deploy

Deploy a Docker Compose application to an Azur...

← Azure App Service deploy ⓘ

Connection type * ⓘ
Azure Resource Manager

Azure subscription * ⓘ
Azure Pass - スポンサー プラン(02955f6...

App Service type * ⓘ
Web App on Windows

App Service name * ⓘ
aaaa23424
bbbb234263456

Package or folder * ⓘ
\$(System.DefaultWorkingDirectory)/**/*.zip

File Transforms & Variable Substitution Op... ▾

Additional Deployment Options ▾

Post Deployment Action ▾

Application and Configuration Settings ▾

About this task

Add

必要事項を記入して「Add」をクリック

```
1 # ASP.NET Core
2 # Build and test ASP.NET Core projects targeting
3 # Add steps that run tests, create a NuGet packa
4 # https://docs.microsoft.com/azure/devops/pipel
5
6 trigger:
7   - main
8
9 pool:
10  - vmImage: ubuntu-latest
11
12 variables:
13  - buildCo
14
15 steps:
16  - script: dotnet build --configuration $(buildCo
17    displayName: dotnet build $(buildConfigurati
18
19 Settings
20 - task: AzureRmWebAppDeployment@4
21   inputs:
22     - ConnectionType: 'AzureRM'
23     - azureSubscription: 'Azure Pass - スポンサー
24     - appType: 'webApp'
25     - WebAppName: 'aaaa23424'
26     - packageForLinux: '$(System.DefaultWorkingDi
```

タスクが追加される

リリース トリガー

パイプラインやステージを自動的に起動する仕組み

※クラシック パイプラインのみ

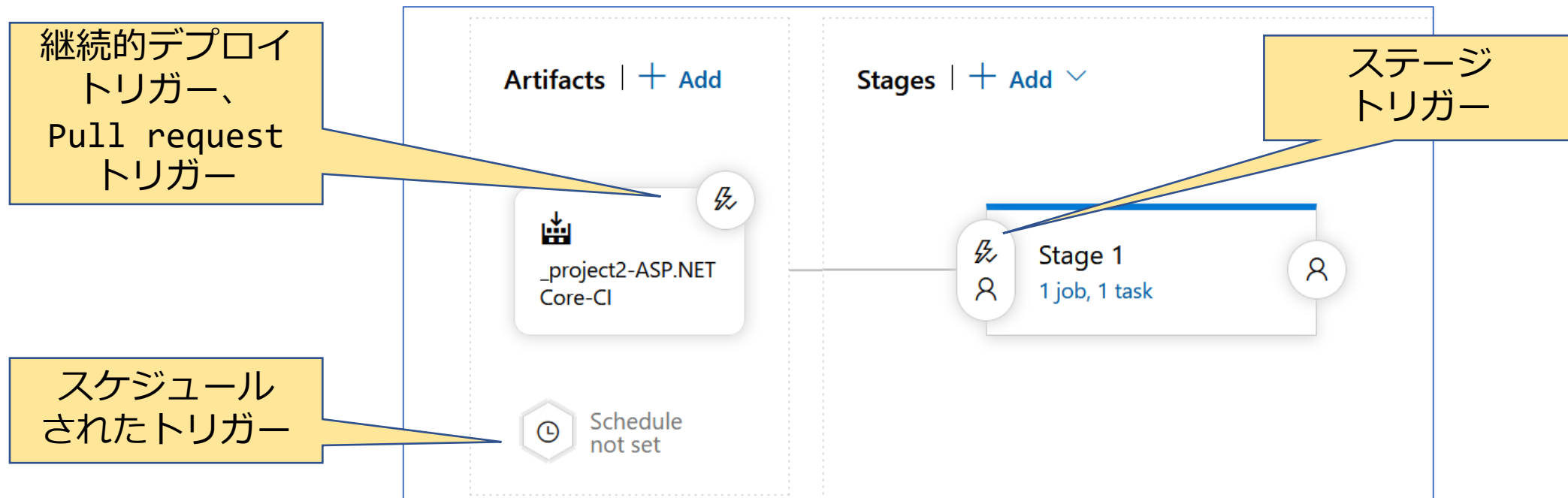
※YAMLパイプラインでは「リリース トリガー」は使用できないが、機能的には YAMLパイプラインの「トリガー」が対応する。

クラシック: <https://learn.microsoft.com/ja-jp/azure/devops/pipelines/release/triggers>

YAML: <https://learn.microsoft.com/ja-jp/azure/devops/pipelines/build/triggers>

リリース トリガーとは？

- 条件が満たされた際に、リリース パイプラインを自動的に起動して、（ビルドパイプラインで作成された）成果物をデプロイする機能。
- リリース パイプラインの設定の一部。



リリース トリガーの例

- 継続的デプロイ トリガー

- 条件: ビルド パイプラインが完了し、成果物が使用可能になった
- 使用例: ビルドパイプラインが完了したら自動でリリースパイプラインを起動したい場合

テスト環境でテスト中にデプロイが行われる可能性があるため、テスト環境には向かない

- スケジュールされたトリガー

- 条件: 指定した曜日・時刻になった
- 使用例: 毎週月曜日の午前9時に成果物をテスト環境へデプロイしてテストしたい場合など

テスト環境に向く

- Pull request トリガー

- 条件: プルリクエスト作成時や変更時に動いたビルド パイプラインが完了し、成果物が使用可能になった
- 使用例: プルリクエスト作成時に、プルリクエストを取り込んだ成果物を生成し、プルリクエスト検証用環境にデプロイして検証したい場合など

プルリク検証環境に向く

- ステージ トリガー: 特定のステージで使用されるトリガー

- 条件: 他のステージで、デプロイが成功した (など)
- 使用例: 「QA環境」ステージを実行し、そのデプロイが成功したら、「本番環境」ステージを実行して本番環境にデプロイする、など

本番環境に向く

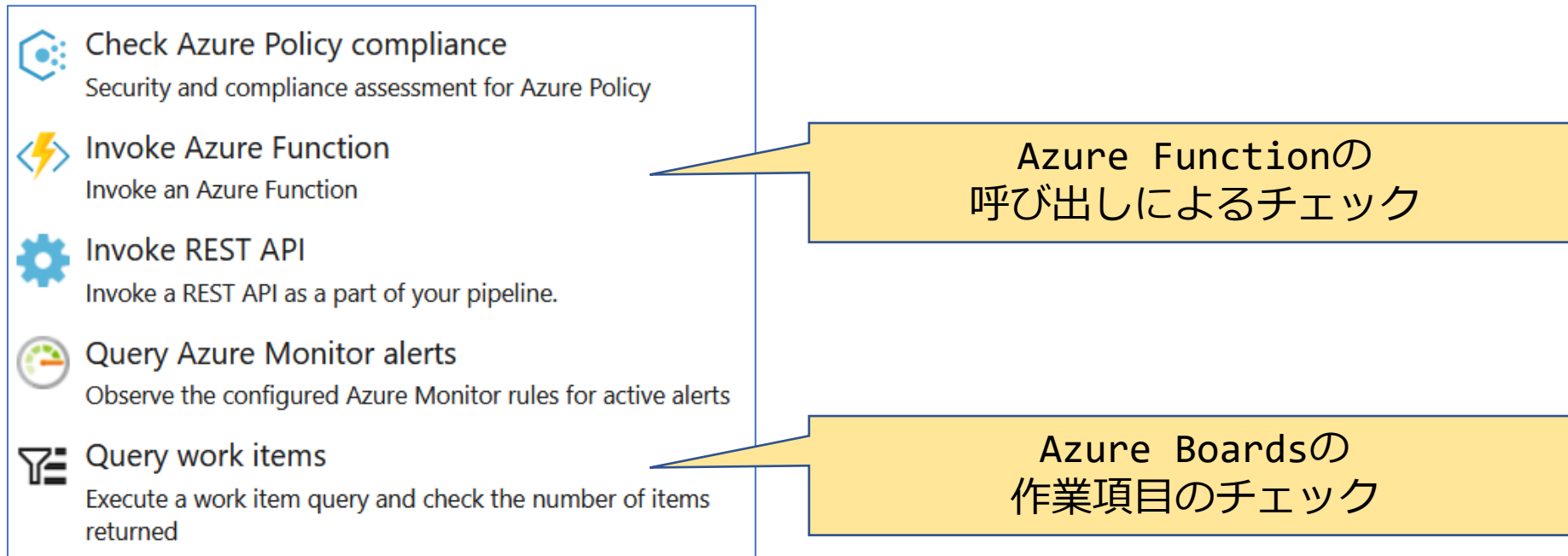
ゲート

状況によって、デプロイ実行やステージ進行をストップさせる機能。
教材等では「リリース ゲート」と書かれている場合もある。

クラシック パイプライン、YAMLパイプライン両方で利用可能。

「ゲート」とは？

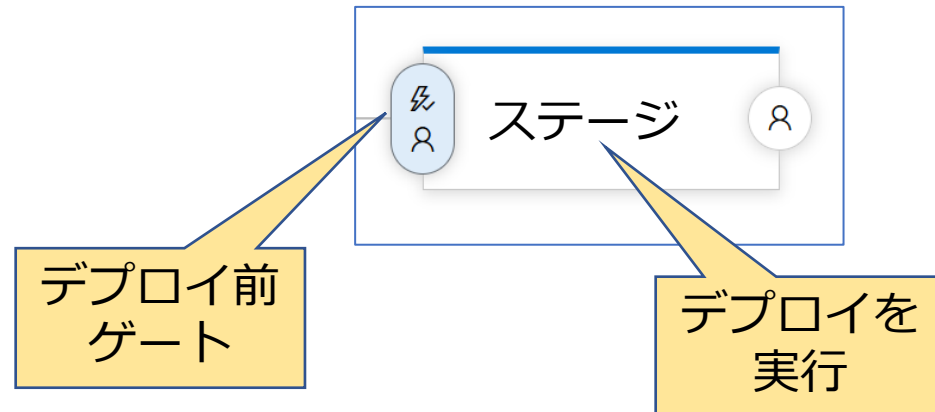
- **機械的な**チェックにより、デプロイ実行やステージ進行をストップさせる機能。



- 「デプロイ前ゲート」と「デプロイ後ゲート」がある。

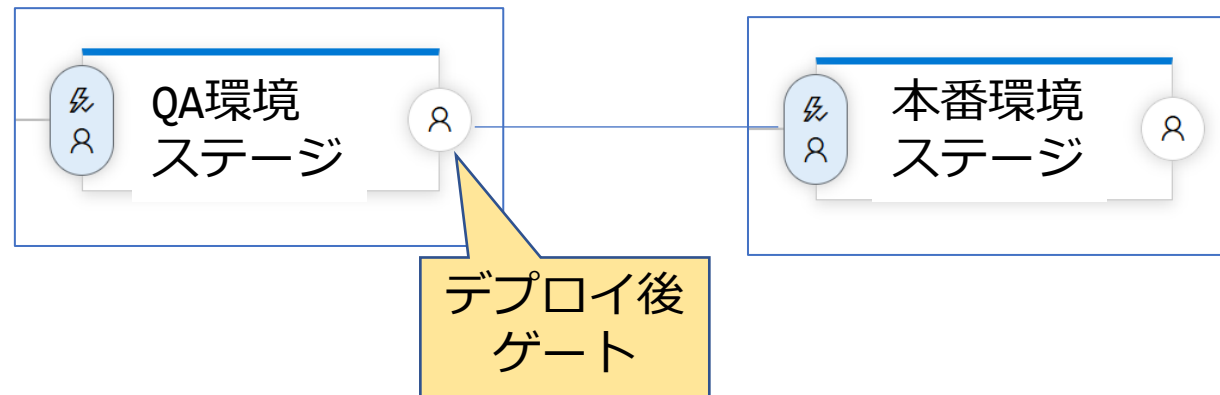
「デプロイ前ゲート」の例

- Azure Boardの作業項目でバグを管理している。
- DONE（完了）になっていないバグが存在するかどうかをゲートで確認する。
- バグがある場合は、デプロイをしない
- バグがない場合は、デプロイを実行



「デプロイ後ゲート」の例

- 「QA環境」ステージと「本番環境」ステージがある。
- まず「QA環境」ステージへのデプロイを実行し、そのデプロイが成功したかどうかをゲートで確認する。
 - 成功なら、次の「本番環境」ステージのデプロイを開始する。
 - 失敗なら、次の「本番環境」ステージのデプロイは開始しない。



■ゲートの設定方法


Pre-deployment conditions


Stage 1

Triggers ^

Define the trigger that will start deployment to this stage

Select trigger ⓘ


After release


Manual only

Artifact filters ⓘ

☐ Disabled

Schedule ⓘ

☐ Disabled

Pull request deployment ⓘ

☐ Disabled

Pre-deployment approvals

Select the users who can approve or reject deployments to this stage

☐ Disabled

Gates

Define gates to evaluate before the deployment.

[Learn more](#)

☐ Disabled

Deployment queue settings v

Define behavior when multiple releases are queued for deployment

Artifacts | + Add



_project2-ASP.NET
Core-CI



Schedule
not set

デプロイ前の
条件設定

デプロイ後の
条件設定



Stage 1
1 job, 1 task



デプロイ後
ゲート

デプロイ前
ゲート

Post-deployment conditions

Stage 1

Post-deployment approvals

Select the users who can approve or reject deployments to this stage

☐ Disabled

Gates

Define gates to evaluate after the deployment.

[Learn more](#)

☐ Disabled

Auto-redeploy trigger

Configure the events that trigger automated redeployment.

☐ Disabled

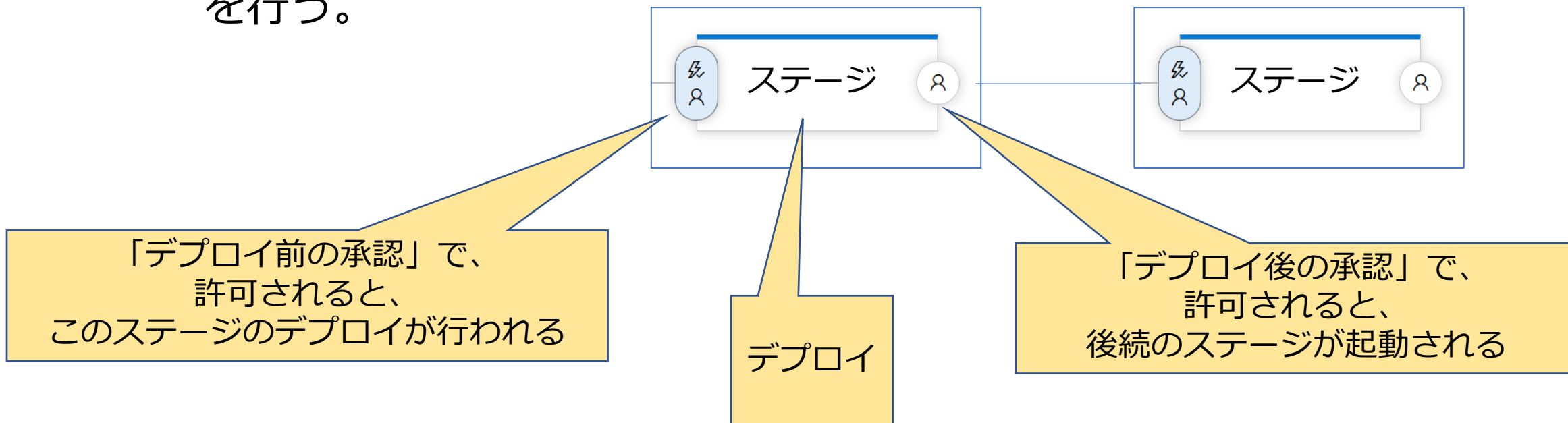
承認

人間による承認（許可または拒否）。

クラシック パイプライン、YAMLパイプライン両方で利用可能。

「承認」とは？

- ゲートと同じだが、デプロイの前または後に**人間が**チェックを行う機能。
- 人間が状況を判断**した上で、approve（許可）または reject（拒否）を行う。



■承認の回答方法

Azure
Pipelines内


Production

Pre-deployment conditions • ⌚ Pending approval

Approvers | ➡ View logs

⌚ Approval pending for 3 minutes ⌚ Timeout in 30d

Waiting for all approvers to approve in sequence .



remotely ask

⌚ Pending for 3 minutes

Reassign

Comment

Checked and OK to release

☐ Defer deployment for later

Approve

Reject

<https://learn.microsoft.com/en-us/azure/devops/pipelines/release/deploy-using-approvals>

Microsoft Azure DevOps

pipelines-example >

Your approval is required for stage 'Deploy' to use environment 'staging'

Requested for

Review approval

Summary

Run reason

Run queued on


Stage requested for

Attempt 1

メール

<https://qiita.com/fukasawah/items/7827df5e645f5bdec992>

Azure Pipelines 10:11 PM



Pre-deployment approval pending for release **Release-12** on stage **PreProd**

Release pipeline: TFS Deployment

Artifacts: 20190306.1

Branch: refs/heads/master

Approvers: Patrick Jane

Approve

Reject

Teams

<https://learn.microsoft.com/ja-jp/azure/devops/pipelines/integrations/microsoft-teams>

承認の設定方法

Pre-deployment conditions

Stage 1

Triggers ^
Define the trigger that will start deployment to this stage

Select trigger ⓘ

After release

Manual only

Artifact filters ⓘ ☐ Disabled

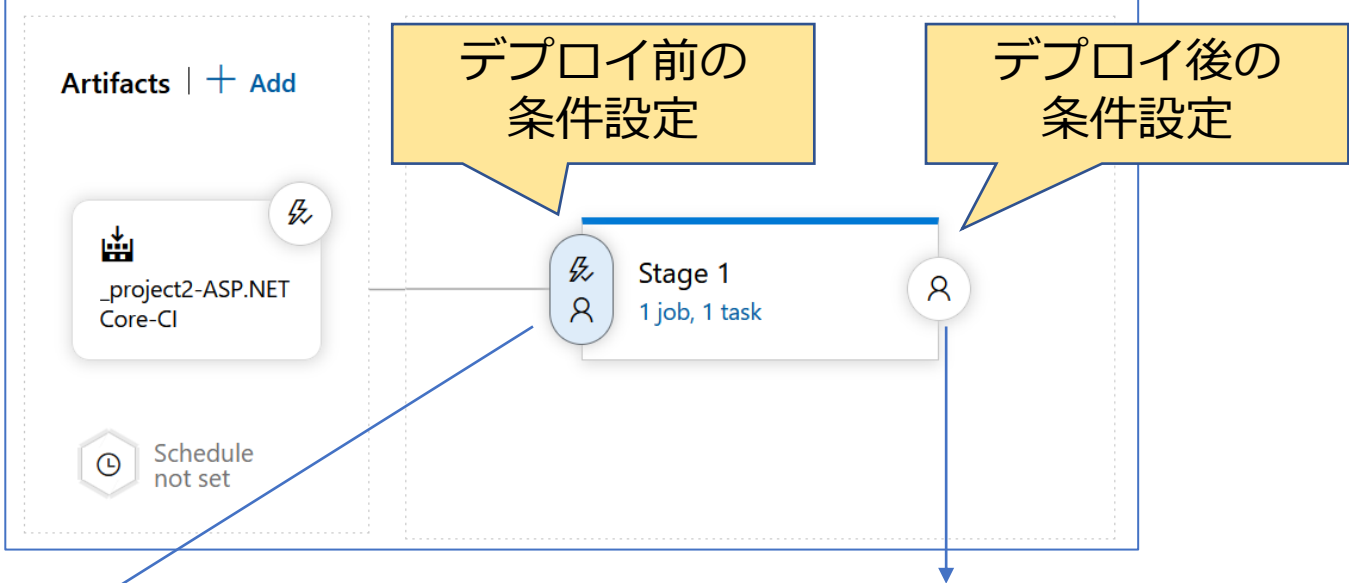
Schedule ⓘ ☐ Disabled

Pull request deployment ⓘ ☐ Disabled

Pre-deployment approvals ⓘ ☐ Disabled
Select the users who can approve or reject deployments to this stage

Gates ⓘ ☐ Disabled
Define gates to evaluate before the deployment.
[Learn more](#)

Deployment queue settings v
Define behavior when multiple releases are queued for deployment



デブロイ後承認

デブロイ前承認

Post-deployment conditions

Stage 1

Post-deployment approvals ⓘ ☐ Disabled
Select the users who can approve or reject deployments to this stage

Gates ⓘ ☐ Disabled
Define gates to evaluate after the deployment.
[Learn more](#)

Auto-redeploy trigger ⓘ ☐ Disabled
Configure the events that trigger automated redeployment.

変数

パイプラインで変数を定義・利用することができる。

クラシック パイプライン、YAMLパイプライン両方で利用可能。

変数とは？

- パイプラインで使用する値を格納する。
- 変数の利点：
 - 値ではなく変数名で書ける（値のハードコーディングを避けることができる）
 - パスワードのような機密情報をYAML内に書かず、変数で参照できる（シークレット変数）
 - 「式」や「組み込み関数」を使用して、パイプライン実行時に変数の値を評価 / 加工できる。
 - `if-else` を使用して、処理中のブランチに応じて値を変えることができる

変数の種類

- ユーザー定義変数：ユーザーが必要に応じて定義する変数
- システム変数（定義済み変数とも）：Azure Pipelinesが自動的に定義する変数
- 環境変数：OS固有の変数

※ 「ユーザー定義変数」と「システム定義変数」は「環境変数」として**も**設定される。

例: ユーザー定義変数 myval => 環境変数 MYVAL

■変数と式の使用例

```
pool:
  vmImage: 'ubuntu-latest'

steps:
- task: PublishPipelineArtifact@1
  inputs:
    targetPath: '$(Pipeline.Workspace)'
    ${{ if eq(variables['Build.SourceBranchName'], 'main') }}:
      artifact: 'prod'
    ${{ else }}:
      artifact: 'dev'
    publishLocation: 'pipeline'
```

ファイル/ディレクトリを
成果物として
発行するタスク

パイプラインの
ワークスペース
ディレクトリ

式

式

処理中の
リポジトリの
ブランチ名

タスクグループ / テンプレート

タスクやステップの集まりに名前をつけて定義する機能。

タスクグループはクラシック パイプラインでのみ利用可能。

テンプレートはYAMLパイプラインでのみ利用可。

タスクグループとは？

- 複数のタスクをグループ化して名前を付ける機能。
- パイプラインのメンテナンス性が向上する。

The screenshot displays the 'SampleApp - 1' pipeline configuration page in Azure DevOps. The 'Tasks' tab is active, showing a list of tasks under the 'Production' deployment process. Two tasks are highlighted: 'Deploy Azure App Service' and 'Cloud Load Test mytestfi...'. A red circle highlights the selection icons (checkmarks) for these tasks. A yellow callout box points to this circle with the text '複数タスクをチェックして選択' (Select multiple tasks by checking). Another yellow callout box points to the 'Create task group' option in the context menu with the text 'タスクグループとして定義！' (Define as task group!). The context menu also includes options like 'Enable selected task(s)', 'Disable selected task(s)', 'Remove selected task(s)', 'Clone task(s)', and 'Manage task group'.

複数タスクを
チェックして選択

タスクグループとして
定義！

テンプレートとは？

- 複数のステップをまとめて別ファイル（テンプレート）にする。
- 親のファイルでテンプレートを参照できる。
- YAMLパイプラインのメンテナンス性が向上する。

```
# File: templates/include-npm-steps.yml
```

```
steps:  
- script: npm install  
- script: yarn install  
- script: npm run compile
```

テンプレートを
定義

テンプレートを
利用

テンプレートを
利用

```
# File: azure-pipelines.yml
```

```
jobs:  
- job: Linux  
  pool:  
    vmImage: 'ubuntu-latest'  
  steps:  
    - template: templates/include-npm-steps.yml  
- job: Windows  
  pool:  
    vmImage: 'windows-latest'  
  steps:  
    - template: templates/include-npm-steps.yml
```