

Azure DevOps との連携

Last updated: 2024.10

1. はじめに

- はじめに
- 実行構成

2. Azure DevOpsの設定

- Azure DevOpsのプロジェクト作成
- リポジトリの作成
- Azure DevOps プロジェクトフォルダーのクローン
- .gitignore ファイルの追加

3. Azure Pipelines agent のセットアップ

4. パイプラインの作成と実行

- パイプラインの作成
- パイプラインの実行

1. はじめに

Ranorex は、Azure DevOps (※) と連携してテストを実行することができます。

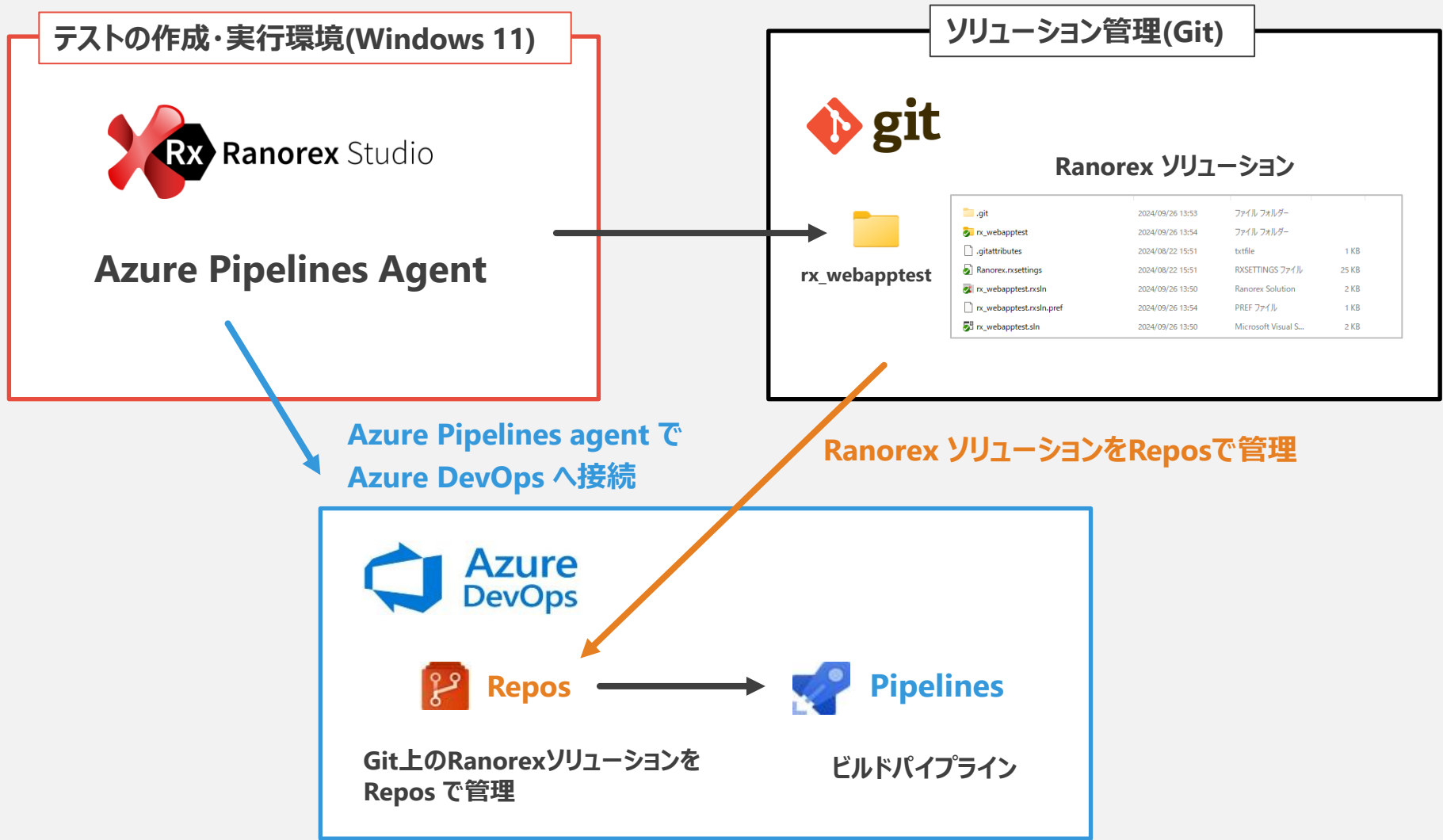
※Microsoft Azure DevOps (ADO) とは、クラウド ベースの CI ソフトウェアです。

Azure DevOps の詳細については、<https://azure.microsoft.com> を参照してください。

本資料では、Ranorex で作成したテストを Azure DevOps のビルドパイプラインで実行する手順について説明します。

Ranorex と Azure DevOps の連携は、以下のRanorex ユーザーガイドでも紹介しています。

<https://support.ranorex.com/ja/userguide/interfaces-connectivity/azure-devops-integration/introduction/>



Azure Pipelines agent で
Azure DevOps へ接続

Ranorex ソリューションをReposで管理

Git上のRanorexソリューションを
Repos で管理

ビルドパイプライン

※本資料では、Ranorex Studio v11.3.3を使用しています

2. Azure DevOps の設定

Azure DevOps プロジェクトを作成します。

Create a project to get started

Project name *
RxDatabaseTest ✓ 1

Description
Database test of Ranorex demo application 2

Visibility

Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.

Private 3
Only people you give access to will be able to view this project.

^ Advanced

Version control ⓘ
Git 4

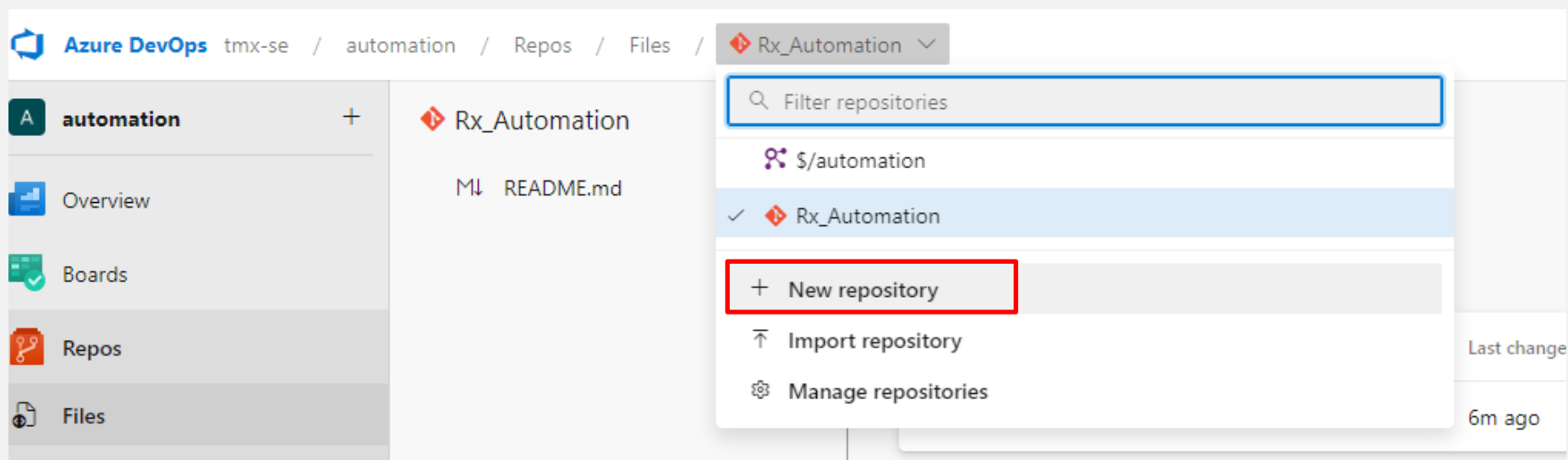
Work item process ⓘ
Basic

+ Create project 3

- 1 プロジェクト名
- 2 プロジェクトの説明 (任意)
- 3 Private (非公開) の設定
- 4 バージョン管理ツールを Git に設定

Git で管理しているRanorex ソリューションを Azure Repos で管理する場合には以下の手順でAzure DevOps 上にリポジトリを作成します。

1. Azure DevOps プロジェクトの左側のメニューから **Repos** を選択します。
2. **New Repository** を選択します。



3. 任意のRepository name を設定のうえ、**Create** を選択します。

Create a repository

Repository type
Git

Repository name *
Rx_Automation

Add a README

Add a .gitignore: None

Your repository will be initialized with a main branch.

Cancel Create

4. リポジトリが生成されます。

Azure DevOps tms-se / automation / Repos / Files / Rx_Automation

automation Rx_Automation

Overview Boards Repos Files Commits Pushes Branches Tags Pull requests Advanced Security Pipelines Test Plans Artifacts

main / Type to find a file or folder.

Set up build Clone

Name	Last change	Commits
README.md	9月10日	eb86b44c Added README.md 聖乃片瀬

Introduction
TODO: Give a short introduction of your project. Let this section explain the objectives or the motivation behind this project.

Getting Started
TODO: Guide users through getting your code up and running on their own system. In this section you can talk about:

1. Installation process
2. Software dependencies
3. Latest releases
4. API references

Build and Test
TODO: Describe and show how to build your code and run the tests.

Contribute
TODO: Explain how other users and developers can contribute to make your code better.

If you want to learn more about creating good readme files then refer the following [guidelines](#). You can also seek inspiration from the below readme files:

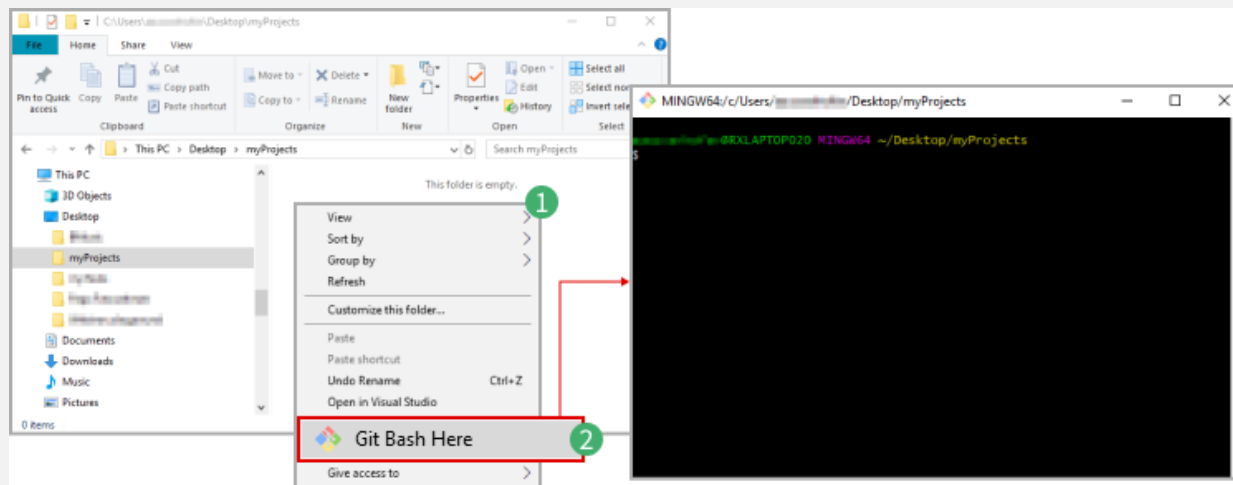
- [ASP.NET Core](#)
- [Visual Studio Code](#)
- [Chakra Core](#)

Azure DevOps プロジェクトフォルダーをリポジトリフォルダーにクローンします。

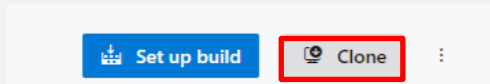


以下の手順でおこないます。

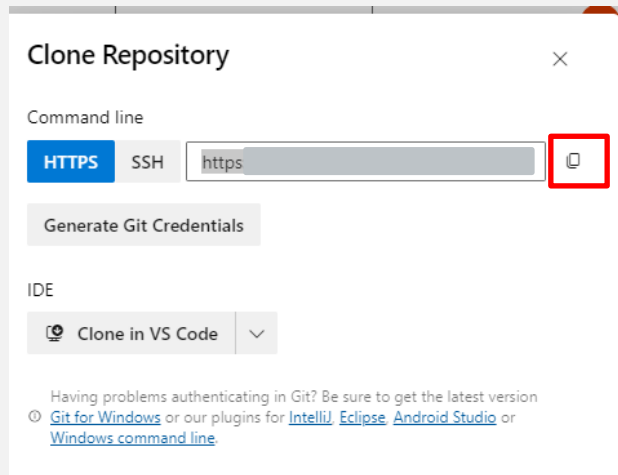
1. 任意の場所にリポジトリフォルダーを作成し、リポジトリフォルダーのなにもないスペースを右クリックします。
2. **Git Bash Here** をクリックしてコマンドラインを開きます。



3. Azure DevOps プロジェクトで、**Repos** をクリックします。
4. 右上の **Clone** をクリックします。



5. **Clone Repository** で、プロジェクトフォルダーのURLをコピーします。



6. コマンドラインに **git clone** と入力し、プロジェクトフォルダーのURLをペーストし、Enterを押下します。



リポジトリフォルダーには、クローンされた Azure DevOps プロジェクトフォルダーが追加されます。

.gitignore ファイルの追加

Git リポジトリの .gitignore ファイルは、特定のファイルをコミットから除外するよう Git に指示します。このファイルを指定せずともインテグレーションをおこなうことができますが、指定しない場合には不要なファイルがコミットに含まれることとなります。

手順については、以下のユーザーガイドをご参照ください。

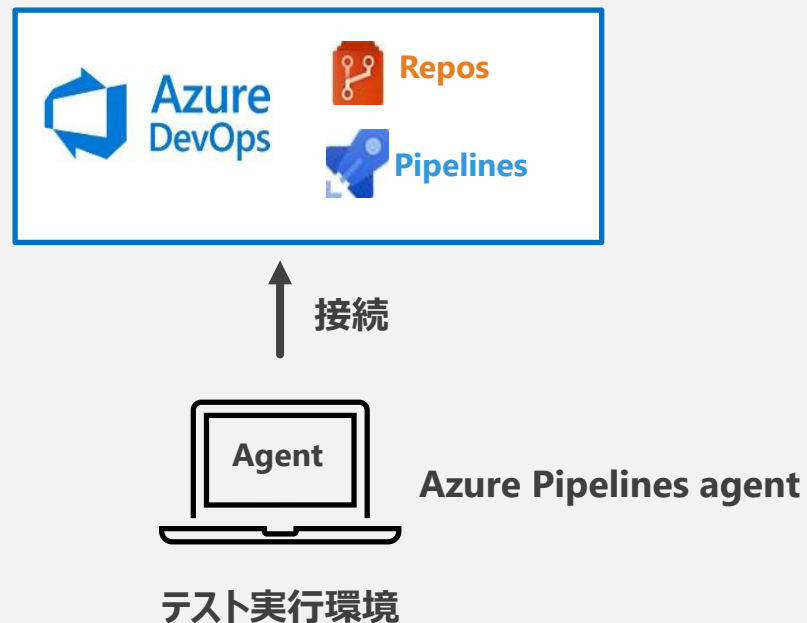
.gitignore ファイルの追加

<https://support.ranorex.com/ja/userguide/interfaces-connectivity/azure-devops-integration/create-ado-project-and-git-repo/#Add.gitignorefile>

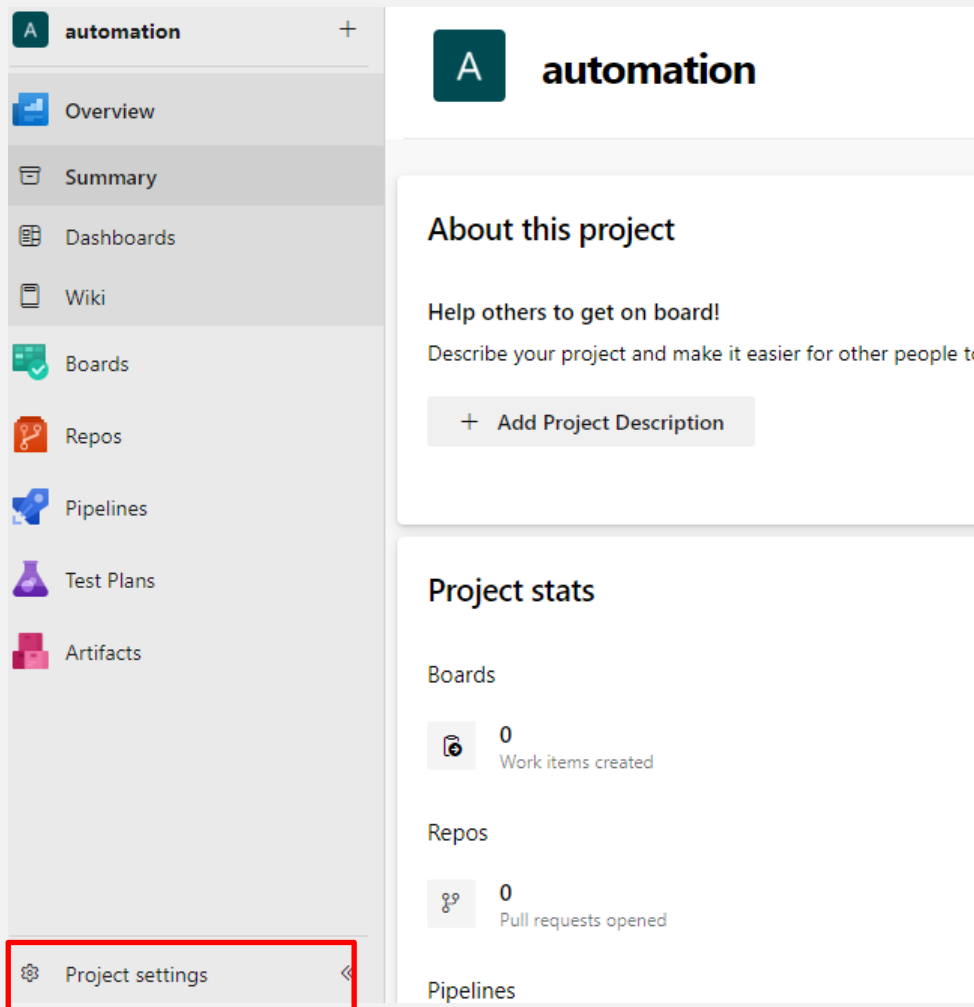
3. Azure Pipelines agent の セットアップ

Azure DevOps パイプラインを実行するには、**Azure Pipelines agent** が必要です。

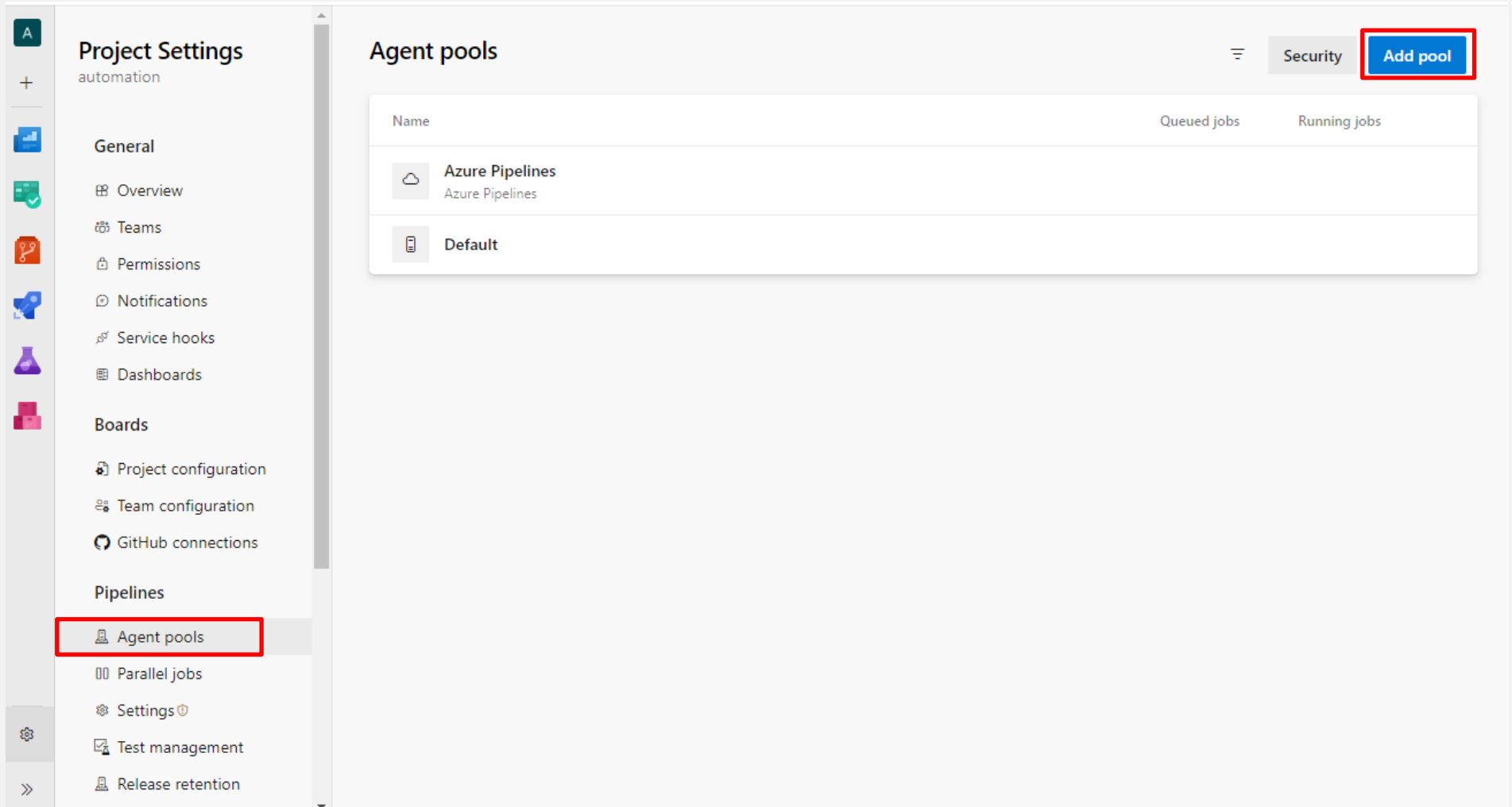
Azure Pipelines agent はテストの実行環境となります。物理マシンまたは仮想マシンに Azure Pipelines agent をインストールし、Azure DevOps に接続すると、パイプラインをエージェントに渡すことができます。





1. Azure DevOps プロジェクトの左下で、**Project settings** をクリックします。



2. Azure pools を選択し、Add pool をクリックします。



The screenshot displays the 'Project Settings' interface for an automation project. The left-hand navigation pane is titled 'Project Settings automation' and lists various configuration categories: General, Boards, Pipelines, and Release retention. The 'Agent pools' option under the 'Pipelines' category is highlighted with a red box. The main content area is titled 'Agent pools' and features a 'Security' tab and an 'Add pool' button, which is also highlighted with a red box. Below the tabs is a table listing existing agent pools.

Name	Queued jobs	Running jobs
 Azure Pipelines Azure Pipelines		
 Default		

3. Azure pools の各種設定をおこない、**Create** ボタンをクリックします。

Add agent pool

Agent pools are shared across an organization.

Pool to link:

New Existing

Pool type:

Self-hosted

A pool of agents that you set up and manage on your own to run jobs. [Learn more.](#)

Name:

Ranorex_Test

Description (optional):

[Markdown supported.](#)

Pipeline permissions:

Grant access permission to all pipelines

Create

Agent pools の一覧にエージェントが追加されます。

Agent pools

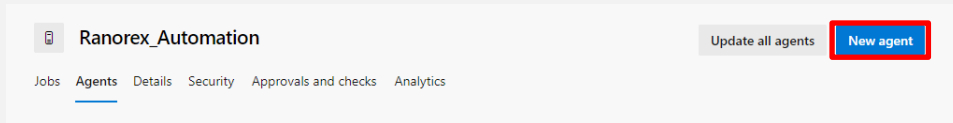


Security

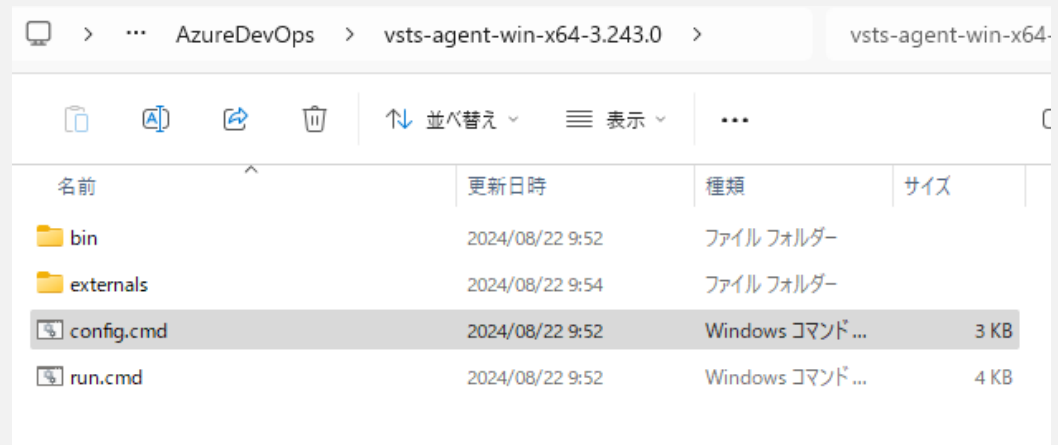
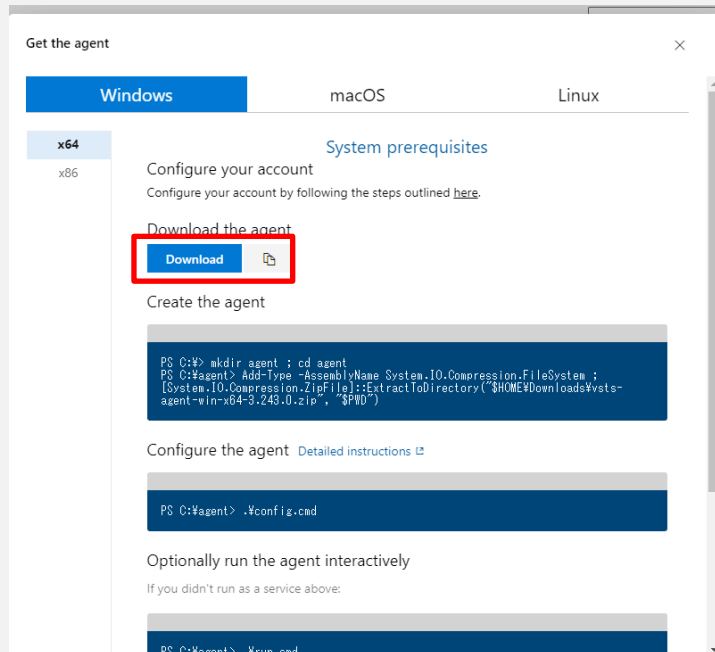
Add pool

Name	Queued jobs	Running jobs
Azure Pipelines Azure Pipelines		
Default		
Ranorex_Automation		

4. 表示されたエージェントビューで、**New agent** をクリックします。

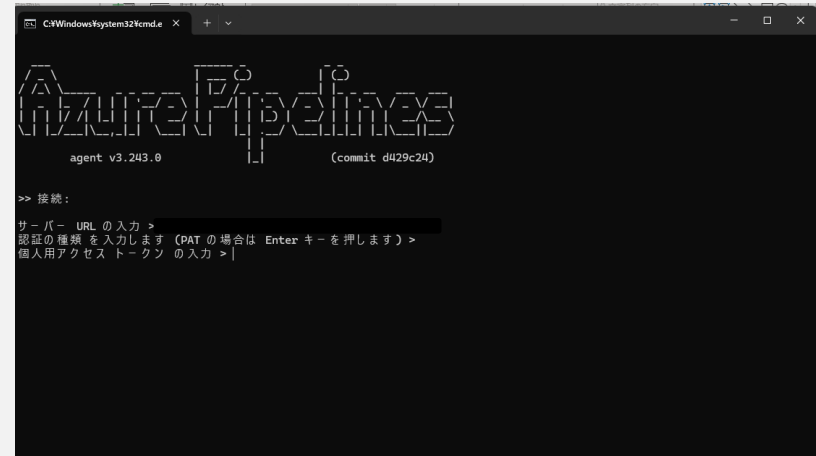
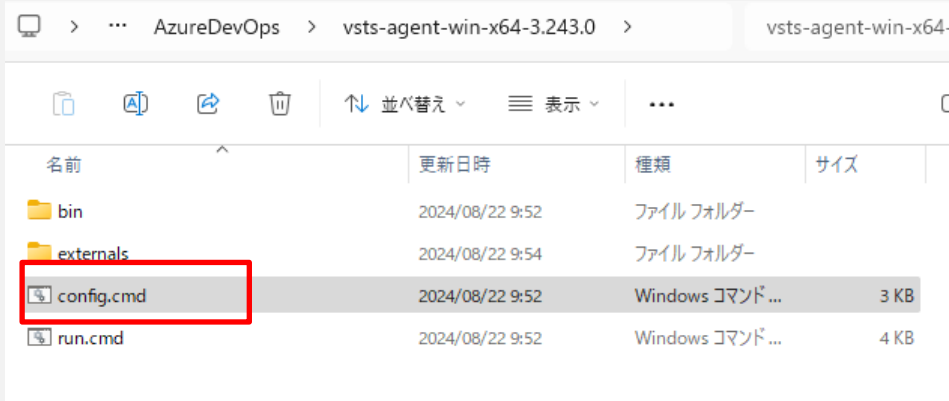


5. OSとしてWindows OSのアーキテクチャ（x64またはx86）を選択し、ダウンロードをクリックします。
6. ダウンロードした.zipファイルを任意のフォルダーに解凍します。



ダウンロードしたエージェントの設定をおこないます。

config.cmd を実行するとコマンドプロンプトが起動し、各種設定がおこなえます。



エージェント設定の詳細は、以下のユーザーガイドをご参照ください。

・エージェントの設定

<https://support.ranorex.com/ja/userguide/interfaces-connectivity/azure-devops-integration/set-up-an-azure-pipelines-agent/#Configuretheagent>

上記ユーザーガイドの 手順5 で記載されているPATの作成方法は、以下のURLをご参照ください。

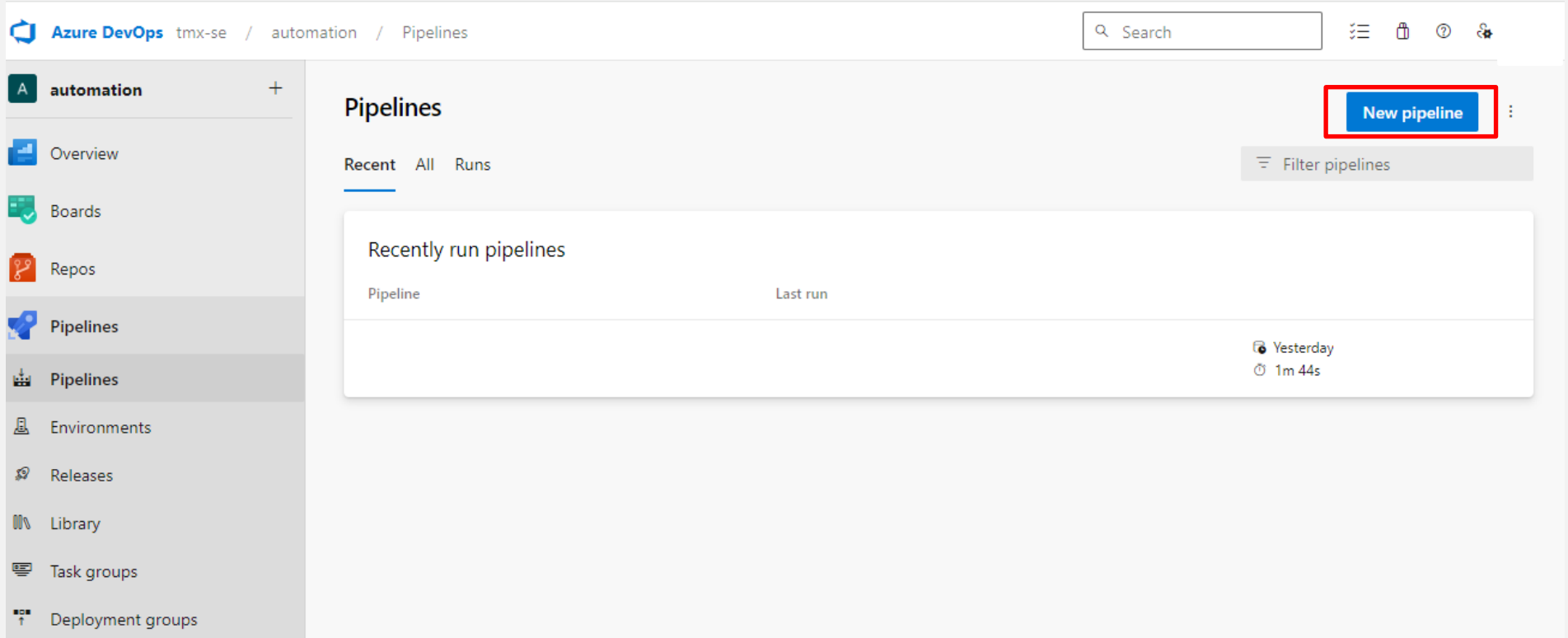
・個人用アクセストークンを使用する

<https://learn.microsoft.com/ja-jp/azure/devops/organizations/accounts/use-personal-access-tokens-to-authenticate?view=azure-devops&tabs=Windows>

4. パイプラインの作成と実行

パイプラインを作成します。

1. プロジェクトの左メニューから **Pipelines** をクリックし、**New pipeline** をクリックします。



The screenshot shows the Azure DevOps interface for the 'Pipelines' section. The left-hand navigation menu is visible, with 'Pipelines' selected. The main content area displays the 'Pipelines' page, which includes a search bar, a 'New pipeline' button (highlighted with a red box), and a 'Filter pipelines' button. Below these, there is a section for 'Recently run pipelines' with a table header showing 'Pipeline' and 'Last run'. The table contains one entry with a clock icon, 'Yesterday', and '1m 44s'.

Azure DevOps tmx-se / automation / Pipelines

Search

automation +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups


Pipelines

Recent All Runs

New pipeline

Filter pipelines

Recently run pipelines

Pipeline	Last run
	 Yesterday 1m 44s

2. Use the classic editor をクリックします。

Azure DevOps tmx-se / automation / Pipelines

Search

automation +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups
- Test Plans
- Artifacts

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
- Team Foundation Version Control**
Centralized version control repositories

Use the classic editor to create a pipeline without YAML.

3. Azure Repos Git を選択し、各種設定をおこないます。

下図では、Ranorex ソリューションを管理している Rx_Automation を Repository として指定しています。

The screenshot shows the Azure DevOps interface for configuring a pipeline source. The left sidebar shows the navigation menu with 'Pipelines' selected. The main area is titled 'Select your repository' and contains a large arrow icon and the text 'Tell us where your sources are. You can customize how to get these sources from the repository later.'

The 'Select a source' section displays several options, with 'Azure Repos Git' highlighted by a red box. Other options include TFVC, GitHub, GitHub Enterprise Server, Subversion, Bitbucket Cloud, and Other Git.

Below the source selection, the 'Team project' dropdown is set to 'automation', the 'Repository' dropdown is set to 'Rx_Automation', and the 'Default branch for manual and scheduled builds' dropdown is set to 'main'. These three dropdowns are also highlighted by a red box. A 'Continue' button is located at the bottom of the configuration area.

4. .NET Desktop を選択し、Apply をクリックします。

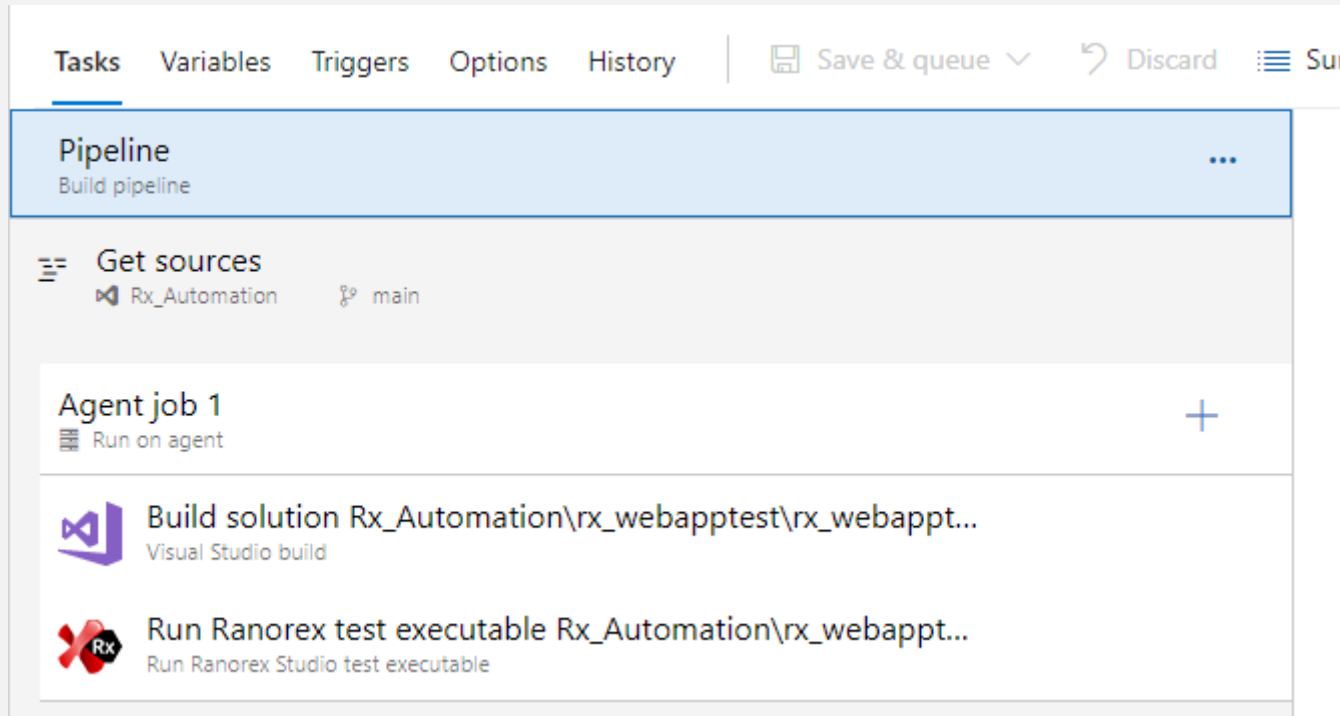
The screenshot shows the Azure DevOps interface for creating a pipeline. The left sidebar contains navigation options: automation, Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area is titled 'Choose a template' and features a large right-pointing arrow icon. Below the icon, it says: 'Choose a template that builds your kind of app. Don't worry if it's not an exact match; you can add and customize the tasks later.'

On the right, the 'Select a template' section includes a search bar and a link to 'Empty job'. Under 'Configuration as code', there is a 'YAML' option with a link to 'Learn more'. The 'Featured' section lists several templates:

- .NET Desktop**: Build and test a .NET or Windows classic desktop solution. The 'Apply' button for this template is highlighted with a red box.
- Android**: Build, test, sign, and align an Android APK.
- ASP.NET**: Build and test an ASP.NET web application.
- Azure Web App for ASP.NET**: Build, package, test, and deploy an ASP.NET Azure Web App.
- Docker container**: Build a Docker image and push it to a container registry.

4. **Agent job** にタスクを追加します。

今回は、**Visual Studio build** と **Run Ranorex Studio test executable** を使用します。



ここで設定したタスクは上から順に実行されます。

タスクを追加する場合には、Agent job 横の + ボタンをクリックします。

Run Ranorex Studio test executable は、検索フォームに Ranorex と入力すると表示されます。

Add をクリックすると、Agent job に追加されます。

The screenshot shows the Azure DevOps Pipelines interface for a pipeline named 'automation-.NET Desktop-CI'. The left sidebar shows the pipeline structure with 'Agent job 1' highlighted. The main area shows the 'Add tasks' section with a search box containing 'Ranorex'. The 'Run Ranorex Studio test executable' task is displayed, and the 'Add' button is highlighted. Below it, the 'Ranorex Azure DevOps build task' is also visible.

Run Ranorex Studio test executable は、Ranorex Studio の公式ビルドタスクとなり、さまざまなオプションをGUI上から設定できます。詳細は、以下のURLをご参照ください。

- ・公式 Ranorex Studio ビルド タスク

<https://support.ranorex.com/ja/userguide/interfaces-connectivity/azure-devops-integration/create-and-execute-an-ado-pipeline/#OfficialRanorexStudiobuildtask>

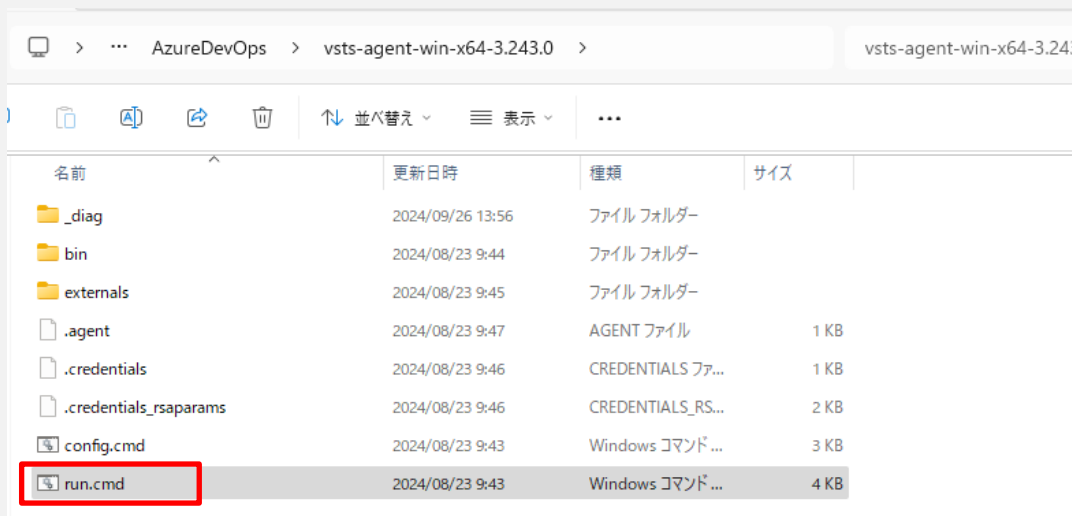
The screenshot displays the Ranorex Automation web interface. The left sidebar shows a pipeline structure with the following tasks:

- Pipeline (Build pipeline)
- Get sources (Run on agent)
- Agent job 1 (Run on agent)
- Build solution Rx_Automation\rx_webapptest\rx_webappt... (Visual Studio build)
- Run Ranorex test executable Rx_Automation\rx_webappt... (Run Ranorex Studio test executable) - This task is selected and highlighted.

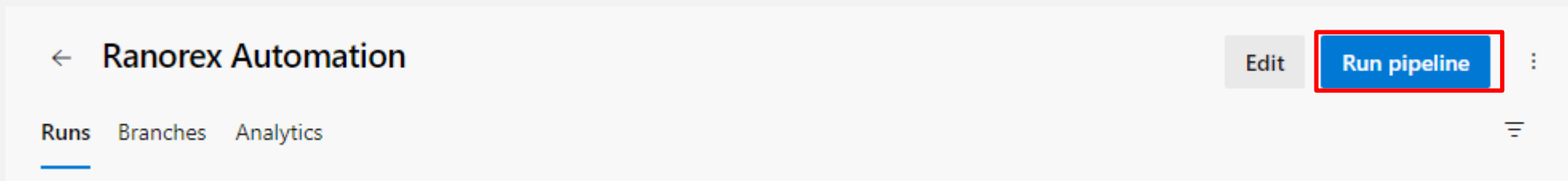
The main panel shows the configuration for the selected task, "Run Ranorex Studio test executable". The configuration includes:

- Task version: 1.*
- Display name: Run Ranorex test executable Rx_Automation\rx_webapptest\rx_webapptest\bin\Debug\rx_webapptest.exe
- General Settings: Test Executable * (Rx_Automation\rx_webapptest\rx_webapptest\bin\Debug\rx_webapptest.exe)
- Ranorex Test Suite: (Empty field)
- Test Run Configuration: (Empty field)
- Reporting Options: Attach Ranorex Report (Report name: %S_%Y%M%D_%T, Report Level: (Default))

パイプラインを実行する前に、Agent を起動します。
ダウンロードしたエージェントフォルダ配下にある **run.cmd** を実行します。



Run pipeline をクリックします。

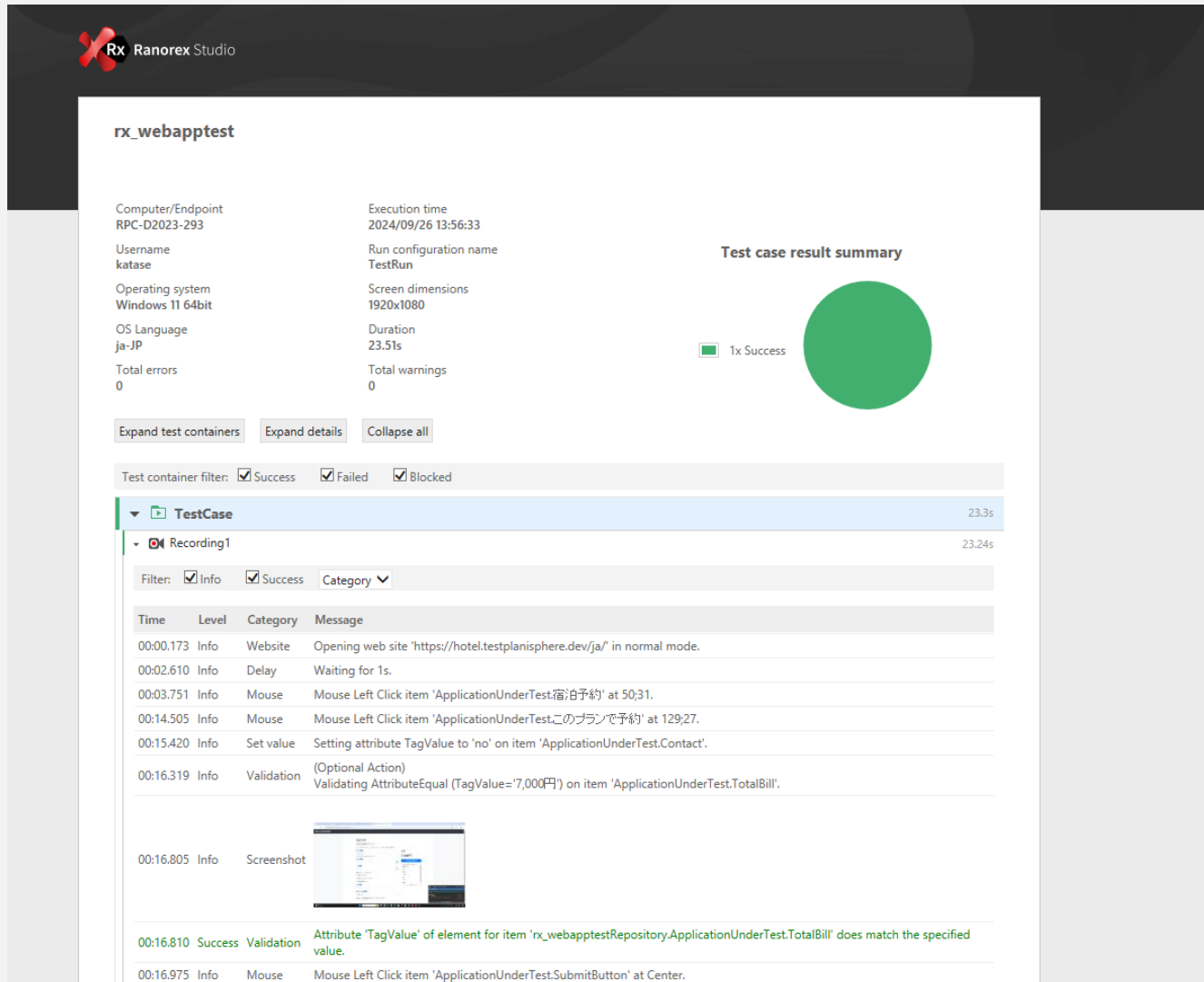


パイプラインが実行されます。
今回のパイプラインでは、ReposからRanorex ソリューションファイルを取得し、
Ranorex ソリューションをビルドした上で、テストが実行されます。

The screenshot displays the Azure DevOps interface for a pipeline run. The left sidebar shows navigation options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area shows the 'Jobs in run #20240926.1' for 'Ranorex Automation'. A table lists the jobs: 'Agent job 1' (1m 22s), 'Initialize job' (<1s), 'Checkout Rx_Automat...' (10s), 'Build solution Rx_Auto...' (14s), 'Run Ranorex test exec...' (55s), 'Post-job: Checkout Rx...' (<1s), 'Finalize Job' (<1s), and 'Report build status' (<1s). The 'Checkout Rx_Automat@main to s' job is selected, showing its log output. The log starts with 'Starting: Checkout Rx_Automation@main to s' and includes task details like 'Task: Get sources', 'Description: Get sources from a repository...', 'Version: 1.0.0', and 'Author: Microsoft'. It then shows the execution of 'git version', 'git lfs version', and 'git config' commands, followed by 'git --config-env=http.extraheader=env_var_http.extraheader fetch --force --tags --prune --prune-tags --progress --no-recurse-submodules origin --depth=1 +0fd158bb13d5272a46b9d5' and 'git --config-env=http.extraheader=env_var_http.extraheader fetch --force --tags --prune --prune-tags --progress --no-recurse-submodules origin --depth=1 +0fd158bb13d5272a46b9d5' commands, both resulting in 'remote: Azure Repos' and 'remote: Found 0 objects to send. (0 ms)'. The log ends with 'remote: Azure Repos'.

```
1 Starting: Checkout Rx_Automation@main to s
2 =====
3 Task      : Get sources
4 Description : Get sources from a repository. Supports Git, TfsVC, and SVN repositories.
5 Version   : 1.0.0
6 Author    : Microsoft
7 Help      : [More Information](https://go.microsoft.com/fwlink/?LinkId=798199)
8 =====
9 Syncing repository: Rx_Automation (Git)
10 Prepending Path environment variable with directory containing 'git.exe'.
11 git version
12 git version 2.45.2.windows.1
13 git lfs version
14 git-lfs/3.4.0 (GitHub; windows amd64; go 1.20.6; git d06d6e9e)
15 git config --get remote.origin.url
16 git config gc.auto 0
17 git config core.longpaths true
18 git config --get-all
19 git config --get-all http.extraheader
20 git config --get-regexp .*extraheader
21 git config --get-all http.proxy
22 git config http.version HTTP/1.1
23 git --config-env=http.extraheader=env_var_http.extraheader fetch --force --tags --prune --prune-tags --progress --no-recurse-submodules origin --depth=1 +0fd158bb13d5272a46b9d5
24 remote: Azure Repos
25 remote:
26 remote: Found 0 objects to send. (0 ms)
27 git --config-env=http.extraheader=env_var_http.extraheader fetch --force --tags --prune --prune-tags --progress --no-recurse-submodules origin --depth=1 +0fd158bb13d5272a46b9d5
28 remote: Azure Repos
```

Ranorex のテストを実行したことで、Ranorex レポートが出力されています。



The screenshot displays the Ranorex Studio interface with a test report for 'rx_webapptest'. The report includes a 'Test case result summary' section with a green circle indicating '1x Success'. Below this, there is a table of test results with columns for Time, Level, Category, and Message. The table shows various actions such as opening a website, waiting, clicking items, and validating attributes. A screenshot of the application under test is also included in the report.

rx_webapptest


Computer/Endpoint: RPC-D2023-293
Execution time: 2024/09/26 13:56:33
Username: katase
Run configuration name: TestRun
Operating system: Windows 11 64bit
Screen dimensions: 1920x1080
OS Language: ja-JP
Duration: 23.51s
Total errors: 0
Total warnings: 0

Test case result summary

1x Success

Expand test containers | Expand details | Collapse all

Test container filter: Success Failed Blocked


Time	Level	Category	Message
00:00.173	Info	Website	Opening web site 'https://hotel.testplanisphere.dev/ja/' in normal mode.
00:02.610	Info	Delay	Waiting for 1s.
00:03.751	Info	Mouse	Mouse Left Click item 'ApplicationUnderTest.宿泊予約' at 50:31.
00:14.505	Info	Mouse	Mouse Left Click item 'ApplicationUnderTest.このプランで予約' at 129:27.
00:15.420	Info	Set value	Setting attribute TagValue to 'no' on item 'ApplicationUnderTest.Contact'.
00:16.319	Info	Validation	(Optional Action) Validating AttributeEqual (TagValue='7,000円') on item 'ApplicationUnderTest.TotalBill'.
00:16.805	Info	Screenshot	
00:16.810	Success	Validation	Attribute 'TagValue' of element for item 'rx_webapptestRepository.ApplicationUnderTest.TotalBill' does match the specified value.
00:16.975	Info	Mouse	Mouse Left Click item 'ApplicationUnderTest.SubmitButton' at Center.

お問い合わせ先

テクマトリックス株式会社

ソフトウェアエンジニアリング事業部

ソフトウェアエンジニアリング営業部

 03-4405-7853

 se-info@techmatrix.co.jp

 <https://www.techmatrix.co.jp/product/quality.html>