

Azure Dev Test Labsでテスト環境を構築してみる！

1. 前提条件

以下の準備が必要です。

- Microsoft Azureアカウント（無料試用版でも可です）
- Visual Studio (**Windows**)
- Visual Studio for Mac (**Mac**)
- Git
- AzCopy
 - **Mac**
 - **Windows-64bit**
- **Microsoft Remote Desktop(Macのみ)**
- 折れない心

2. ざっくりとした流れ

1. Azure devtest labs作る
2. 匿名ユーザーでも招待出来るようにVM作成(windows server)
3. VMが出来たら、 RDPに入る
4. **IIS**インストール
5. **.NET Core**ホスティングバンドルのインストール
6. サブネットにネットワークセキュリティグループ割当
7. ネットワークインターフェースとの関連付け
8. Visual StudioからVMにBlazorアプリのデプロイ MacはKuduで公開用のビルドファイルを手動アップ
9. ブラウザで動作確認
10. 作成したVMをスナップショット
11. AzCopy を使用して VHD ファイルをアップロード

3. 手順

3.1. DevTest Labs作成

Azure Portalに入り、検索ボックスからDevTest Labsのサービスを検索し、選択します。

The screenshot shows the Microsoft Azure portal's search interface. The search bar at the top contains the text 'Dev'. Below the search bar, there are two main sections: 'サービス' (Services) and 'リソース' (Resources). In the 'サービス' section, 'DevTest Labs' is listed and has a red box drawn around it. Other services listed include DevOps Starter, Azure Device Updates, Azure DevOps organizations, Classic Dev Services, IoT Hub, and various Windows and Intune-related services. In the 'リソース' section, there are several storage accounts, DevTest Labs, key vaults, and virtual network resources. The bottom of the screen shows a navigation bar with '移動' (Move) selected.

選択したらDevTest Labsのサービスへ移動するので、追加のボタンをクリックします。

The screenshot shows the 'DevTest Labs' service page in the Azure portal. At the top, there is a breadcrumb trail showing 'Home > DevTest Labs'. Below the title, there is a '既定のディレクトリ' (Default Directory) section. The main area features a toolbar with buttons for '+追加' (Add), 'ビューアの管理' (View Management), '更新' (Update), 'CSVにエクスポート' (Export to CSV), 'クエリを開く' (Open Query), 'タグの割り当て' (Assign Tag), and 'フィードバック' (Feedback). Below the toolbar, there are several filter options: '名前でフィルター...' (Filter by Name...), 'サブスクリプション == すべて' (Subscription == All), 'リソース グループ == すべて' (Resource Group == All), '場所 == すべて' (Location == All), and 'フィルターの追加' (Add Filter). The '+追加' button is highlighted with a red box.

DevTestのラボ作成画面が表示されるので基本設定情報を入力していきます。

項目名	入力値
サブスクリプション	無料枠での使用であれば無料試用版と出ていると思うのでそのままOKです。 無料試用版でない方はご自身のサブスクリプションを選択してください。
リソースグループ	何も項目がないと思うので、新規作成のリンクをクリックし、作成してください。 命名は自由ではありますが、リソースグループはrg-で始まるのがMicrosoftの推奨のことです。 命名規則を気にされる方はこちらにある ResourceGroup を参照いただければと思います。
ラボ名	こちらは入力チェックに引っかからないように自由に決めてOKです。
場所	東日本を選択しましょう。

Microsoft Azure

ホーム > DevTest Labs >

Devtest ラボの作成

*基本設定 自動シャットダウン ネットワーク タグ 確認および作成

Devtest Labs を使用すると、開発チームは、仮想マシン (VM) と Azure Resource Manager (ARM) ベースのリソースを効率的に自己管理しながら、中央管理者によって定義されたセキュリティ ポリシーとガードレールを維持できます。Azure DevTest Labs に関する詳細情報をご覧ください。 [Learn more on Azure DevTest Labs](#)

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション * tanabebe-dev-001

リソース グループ * (新規) rg-devtest-labs-001
新規作成

ラボ名 * devtest-labs-env-001

場所 * 東日本

● ラボ ポリシーはラボ環境に適用されません。詳細については、ここをクリックしてください。

パブリックな環境 ① オン オフ

確認および作成 次へ: 自動シャットダウン Automation のテンプレートをダウンロードする

自動シャットダウンを選択して、入力していきます。

項目名	入力値
有効	リソース削除などをしていない場合のコスト管理としても事故らないように、 オン にしておきましょう。
スケジュールされたシャットダウン	画像では19:00で設定していますが、meetupの時間を考慮して 21:00 で設定しましょう。
タイムゾーン	(UTC+09:00) Osaka, Sapporo, Tokyoを選択しましょう。

項目名	入力値
自動シャットダウンの前に通知を送信しますか？	不要なので今回はいいえにします

Microsoft Azure

ホーム > DevTest Labs >

Devtest ラボの作成

* 基本設定 **自動シャットダウン** ネットワーク タグ 確認および作成

Auto-shutdown allows you to automatically shut down all machines in a lab at a scheduled time each day

有効 オン オフ

スケジュールされたシャットダウン

タイムゾーン

自動シャットダウンの前に通知を送信しますか? はい いいえ

確認および作成 前へ: 基本設定 次へ: ネットワーク Automation のテンプレートをダウンロードする

ここまで入力が完了したら**確認および作成**のボタンをクリックしましょう。Azureが検証をしてくれるので問題なければ**作成**のボタンをクリックします。

Microsoft Azure

ホーム > DevTest Labs >

Devtest ラボの作成

成功

* 基本設定 自動シャットダウン ネットワーク タグ 確認および作成

概要

基本設定

名前	devtest-labs-env-001
サブスクリプション名	tanabebe-dev-001
リソース グループ	rg-devtest-labs-001
場所	東日本
パブリックな環境	Enabled

自動シャットダウン

有効	Enabled
スケジュール	Thu Sep 24 2020
タイム ゾーン	Tokyo Standard Time

ネットワーク

Virtual Network	既定値
サブネット	既定値
ラボ リソースの分離	Disabled

タグ

作成 前へ: タグ Automation のテンプレートをダウンロードする

サービスのデプロイが始まるので、少し待ちましょう。完了すると以下のとおりに告知されるのでリソースに移動で移動しましょう。



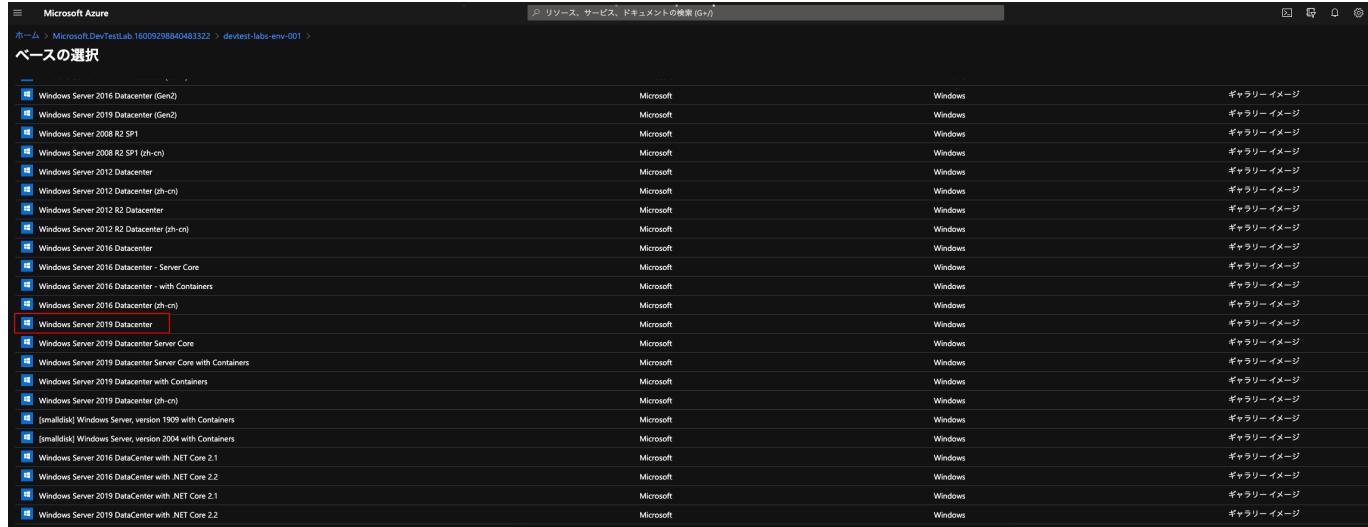
もちろんこの画面を離れても問題ないです。その場合はAzure Portalの通知からデプロイの進行状態が通知から確認できますので、こちらからでもOKです。

DevTest Labsの作成はこれで完了です。次はVMの作成を行います。

3.2. DevTest LabsにVMを作成

DevTest Labs内で追加をクリックします。

ベースの選択が表示されるので**Windows Server 2019 Datacenter**を選択します。



選択するとラボリソースの作成画面が表示されるので基本設定情報を入力していきます。

※ユーザー名とパスワードは忘れないように控えておいて下さい。

項目名	入力値
仮想マシン名	命名は自由ではありますが、VMはvmsで始まるのがMicrosoft推奨となります。
ユーザー名	自由入力となります。
保存されているシークレットを試用する	未チェックとします。
パスワード	自由入力となります。
規定のパスワードとして保存する	チェックします。
ディスクとサイズ	A3が標準で選択されますが、高いのでVMサイズを変更します。サイズの変更リンクをクリックするとサイズが表示されるのでB1msを選択します。
OSディスクの種類	Standard HDDでOKです。

Microsoft Azure

ホーム > Microsoft.DevTestLab.16009298840483322 > devtest-labs-env-001 > ベースの選択 >

ラボ リソースの作成

仮想マシン

基本設定 詳細設定

ユーザー設定

仮想マシン名 ✓

ユーザー名 * ✓

保存されているシークレットを使用する

パスワード * ✓

既定のパスワードとして保存する

ディスクとサイズ

仮想マシン サイズ * ⓘ
Standard_B1ms
サイズの変更

OS ディスクの種類 ⓘ

成果物

成果物の選択
0 個の成果物が選択されています
成果物を追加または削除

作成

自動シャットダウンを選択して、入力していきます。

項目名	入力値
仮想ネットワーク	自動で選択されるので、そのままOKです
サブネットセレクター	自動で選択されるので、そのままOKです
IPアドレス	パブリックにします
仮想マシンの有効期限	未入力でOKです
このマシンを要求可能にする	はいを選択します
インスタンス数	1

Microsoft Azure

ホーム > Microsoft.DevTestLab.16009298840483322 > devtest-labs-env-001 > ベースの選択 >

ラボ リソースの作成

仮想マシン

基本設定 詳細設定

ネットワークの設定

仮想ネットワーク * DtIdevtest-labs-env-001

サブネット セレクター * DtIdevtest-labs-env-001Subnet

IP アドレス **パブリック プライベート 共有**

仮想マシンの有効期限

有効期限 ⓘ 有効期限はありません 日本標準時

要求オプション

このマシンを要求可能にする ⓘ **はい いいえ**

インスタンス数 ⓘ 1

Automation

ARM テンプレートの表示

入力が完了したら基本設定を選択して**作成**ボタンをクリックしてください。デプロイが走りますので待ちましょう。VM作成は時間がかかるので10分ほど... 完了したらDevTest Labにて以下のように要求可能な仮想マシンが現れますので次へ進みます。

Microsoft Azure

リソース、サービス、ドキュメントの検索 (G+ /)

ホーム > DevTest Labs > devtest-labs-env-001 DevTest Lab

概要

リソース グループ (変更) : rg-devtest-labs-001
状態 : 準備完了
場所 : 東日本
サブスクリプション (変更) : tanabebe-dev-001
サブスクリプション ID : [REDACTED]

自分の仮想マシン

名前	状態	自動開始	自動シャットダウン	ベース
表示するものはありません				

要求可能な仮想マシン

名前	状態	自動開始	自動シャットダウン	ベース
vm-lab-wsvr-001	使用可能	いいえ	はい	Windows Server 2019 Datacenter

3.3. VMに接続する

以下のとおり、仮想マシンの要求を行います

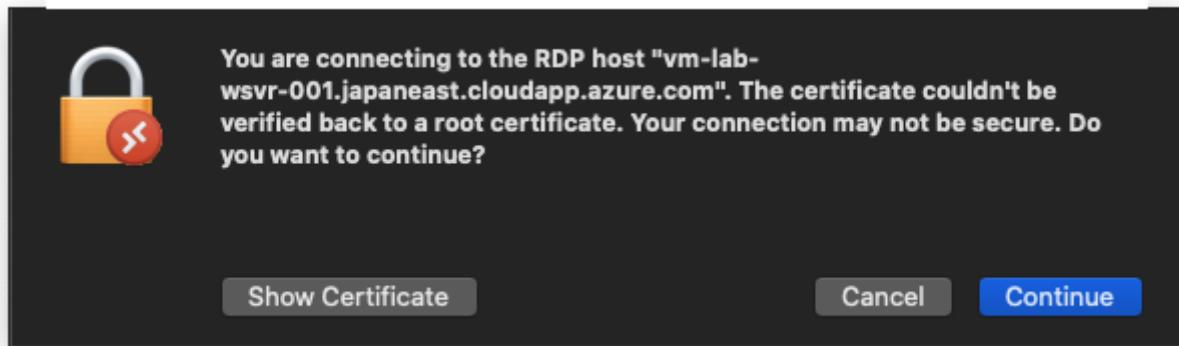
The screenshot shows the Microsoft Azure DevTest Labs interface. In the center, there is a table with one row for a VM named 'vm-lab-wsvr-001'. The columns are 'Name' (名前), 'Status' (状態), 'Automatic Start' (自動開始), 'Automatic Shutdown' (自動シャットダウン), and 'Base' (ベース). The 'Status' column shows 'Available' (使用可能) with a green checkmark. The 'Automatic Shutdown' column has a radio button set to 'Yes' (はい). The 'Base' column shows 'Windows Server 2019 Datacenter' with a green checkmark. On the far right, there is a 'Machine Request' context menu with options: 'Delete' (削除) and 'Edit' (編集). The 'Edit' option is highlighted with a red box.

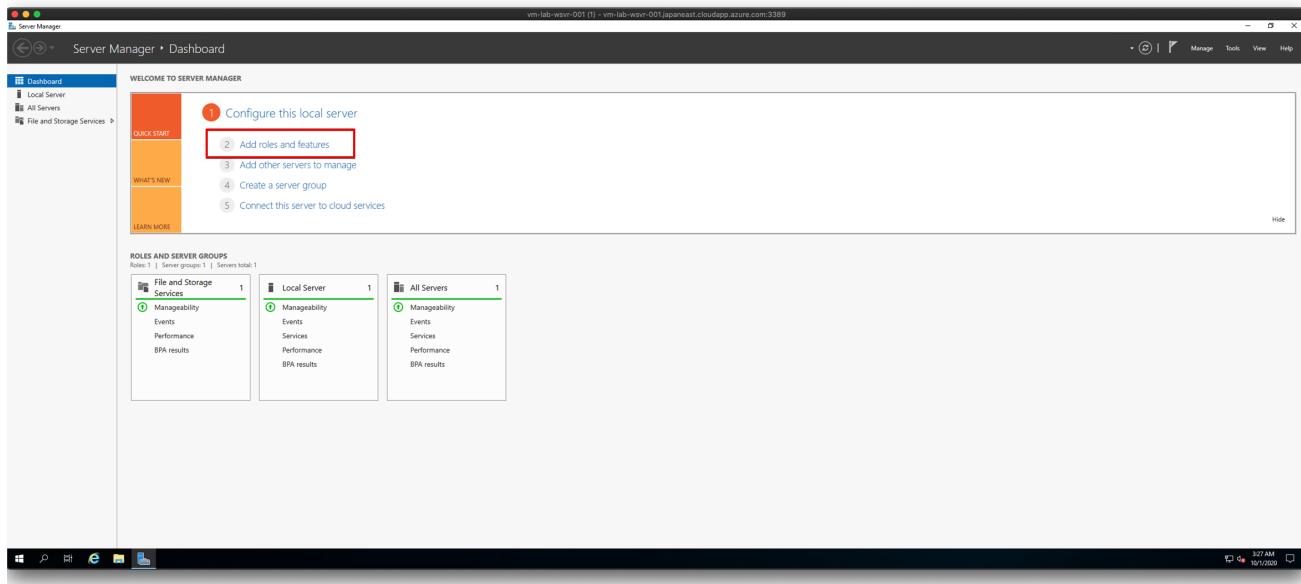
要求が完了すると自分の仮想マシンとしてVMが表示されるので、接続を選択しRDPファイルをダウンロードします。

This screenshot is similar to the previous one, showing the same table and context menu. However, the 'Connect' option in the 'Machine Request' menu is now highlighted with a red box.

ダウンロードした .rdp ファイルをダブルクリックして仮想マシンに接続します。

Macだと以下のようにになりますが、 **Continue** を選択してOKです。





インストールウィザードが立ち上がるるので、ここは**Next**で進みます。

Add Roles and Features Wizard

Before you begin

DESTINATION SERVER
vm-lab-ws01

Before You Begin

- Installation Type
- Server Selection
- Server Roles
- Features
- Confirmation
- Results

This wizard helps you install roles, role services, or features. You determine which roles, role services, or features to install based on the computing needs of your organization, such as sharing documents, or hosting a website.

To remove roles, role services, or features:
Start the Remove Roles and Features Wizard

Before you continue, verify that the following tasks have been completed:

- * The Administrator account has a strong password
- * Network settings, such as static IP addresses, are configured
- * The most current security updates from Windows Update are installed

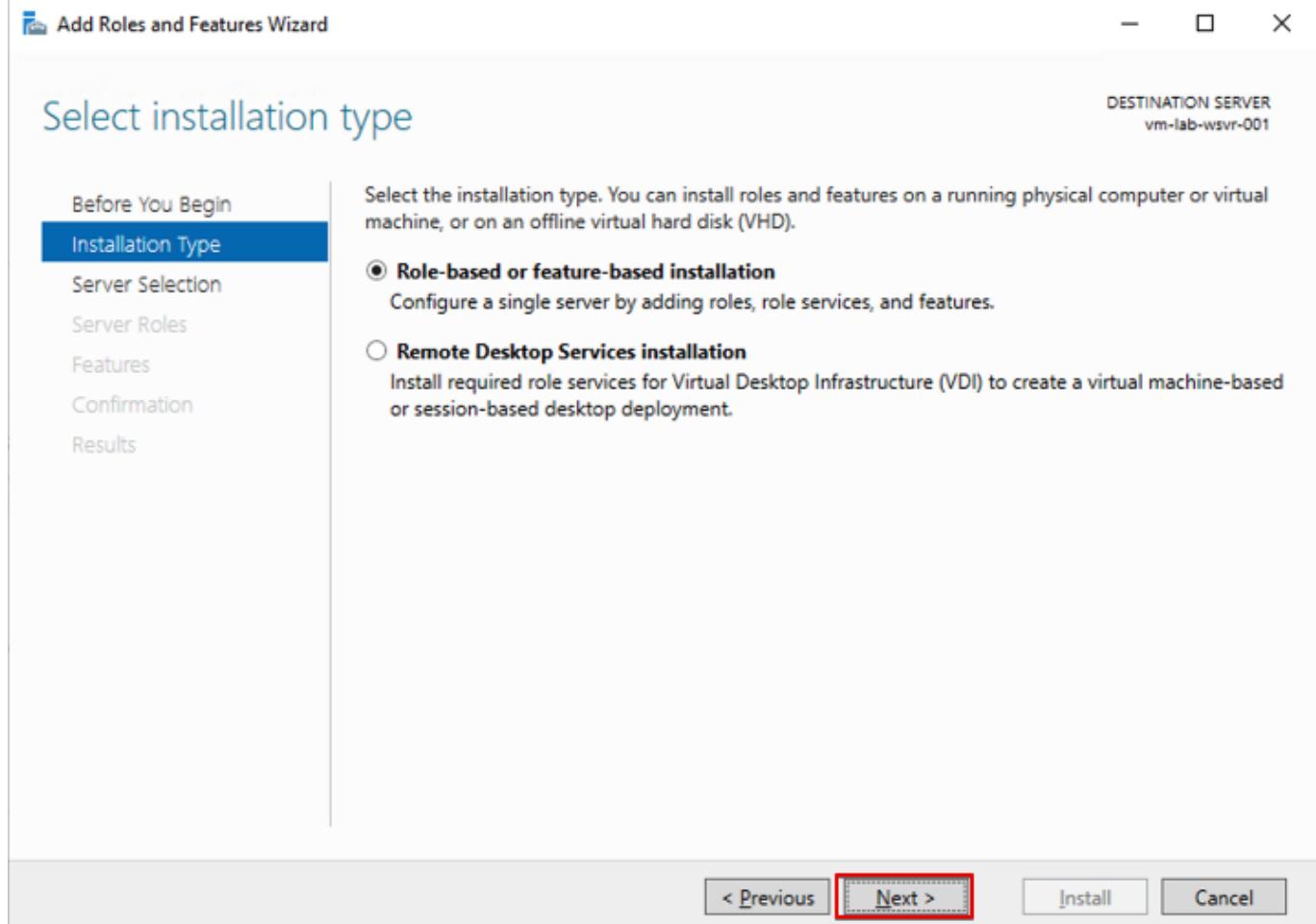
If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.

To continue, click Next.

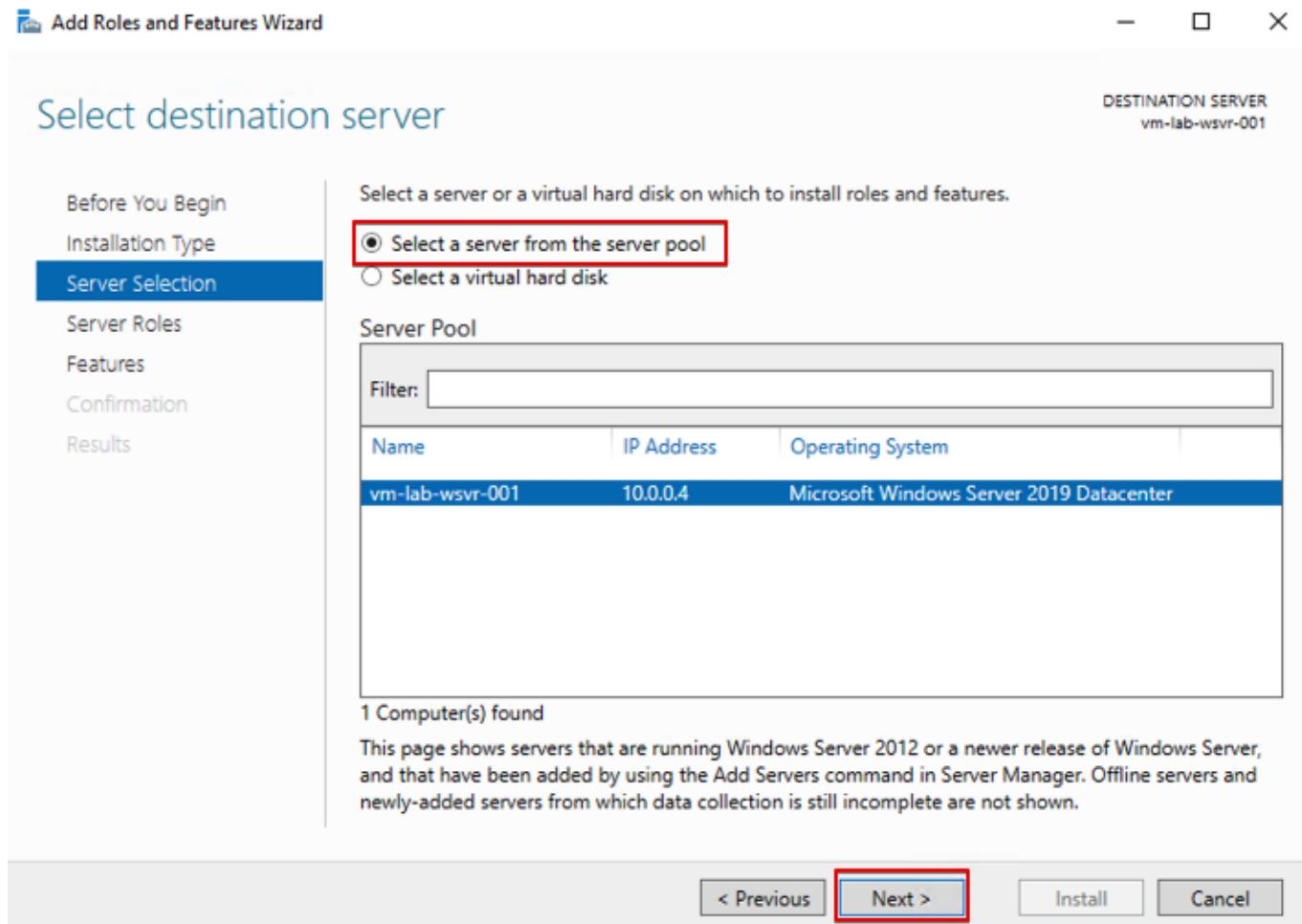
Skip this page by default

[< Previous](#) **Next >** [Install](#) [Cancel](#)

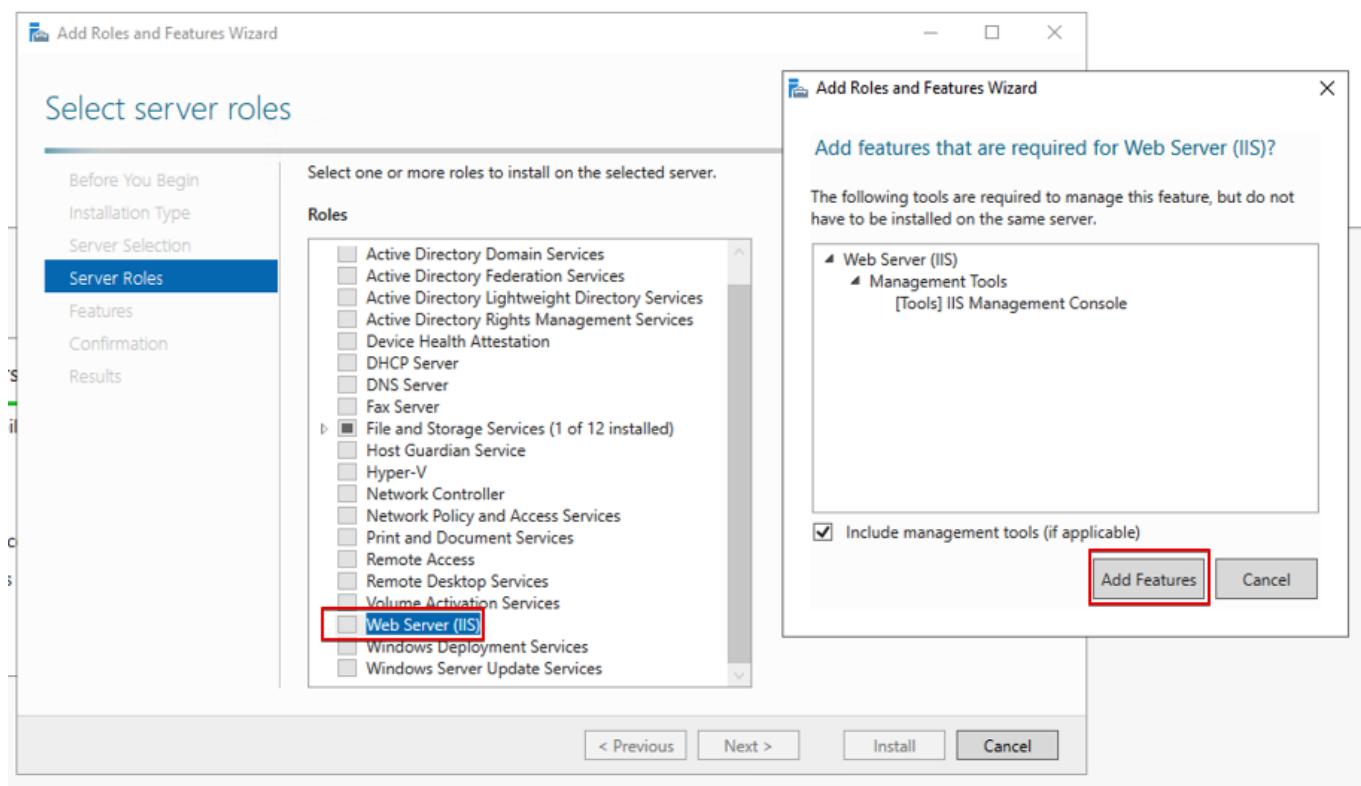
ここも変更せず**Next**で進みます。



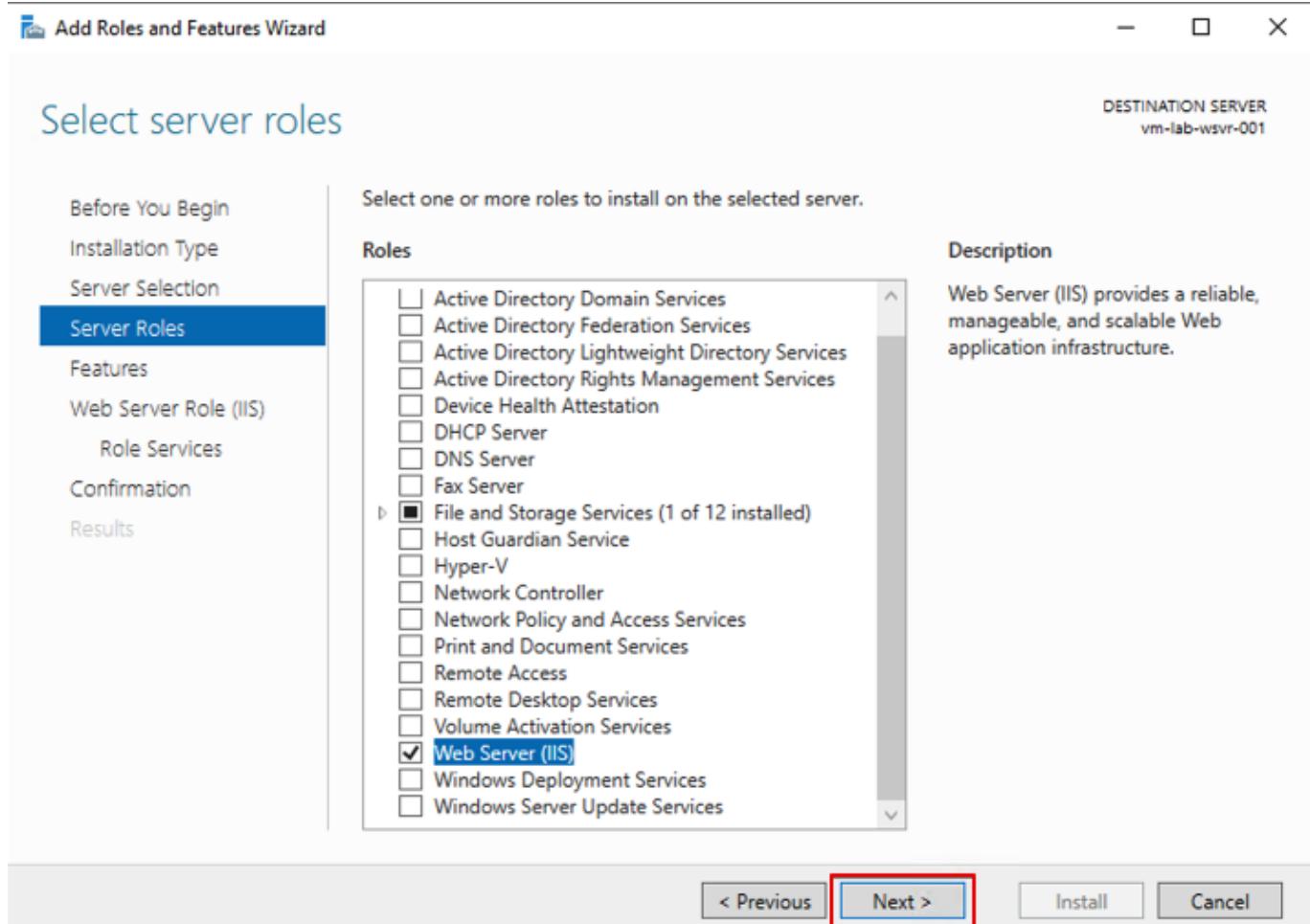
ここも変更せず、 **Next**で進みます。



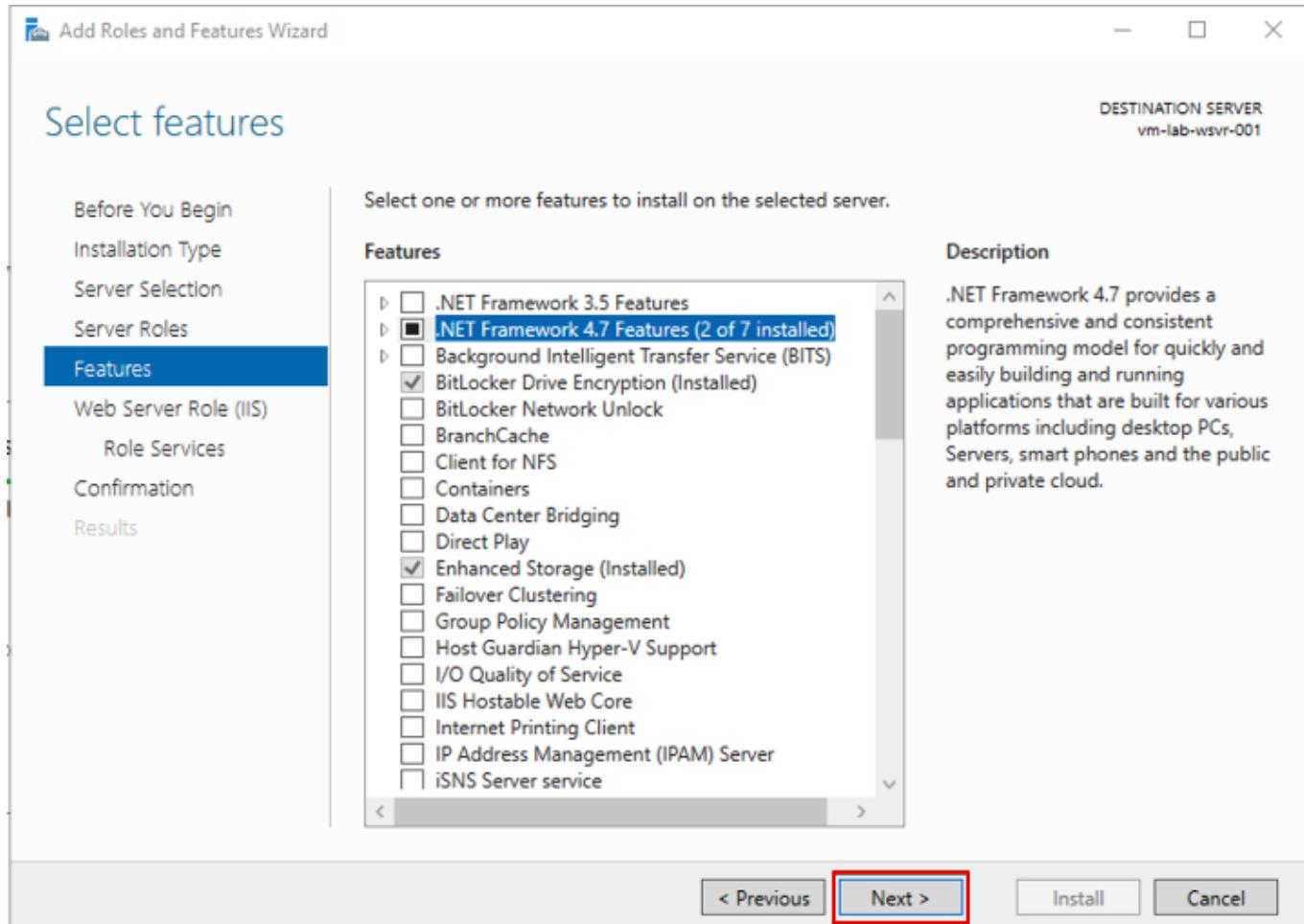
Selectr server rolesまで来たら **Roles**内の**Web Server(IIS)**をクリックします。
クリックするとダイアログが表示されますが、 **Add Features**を選択してください。（以降も同じく）



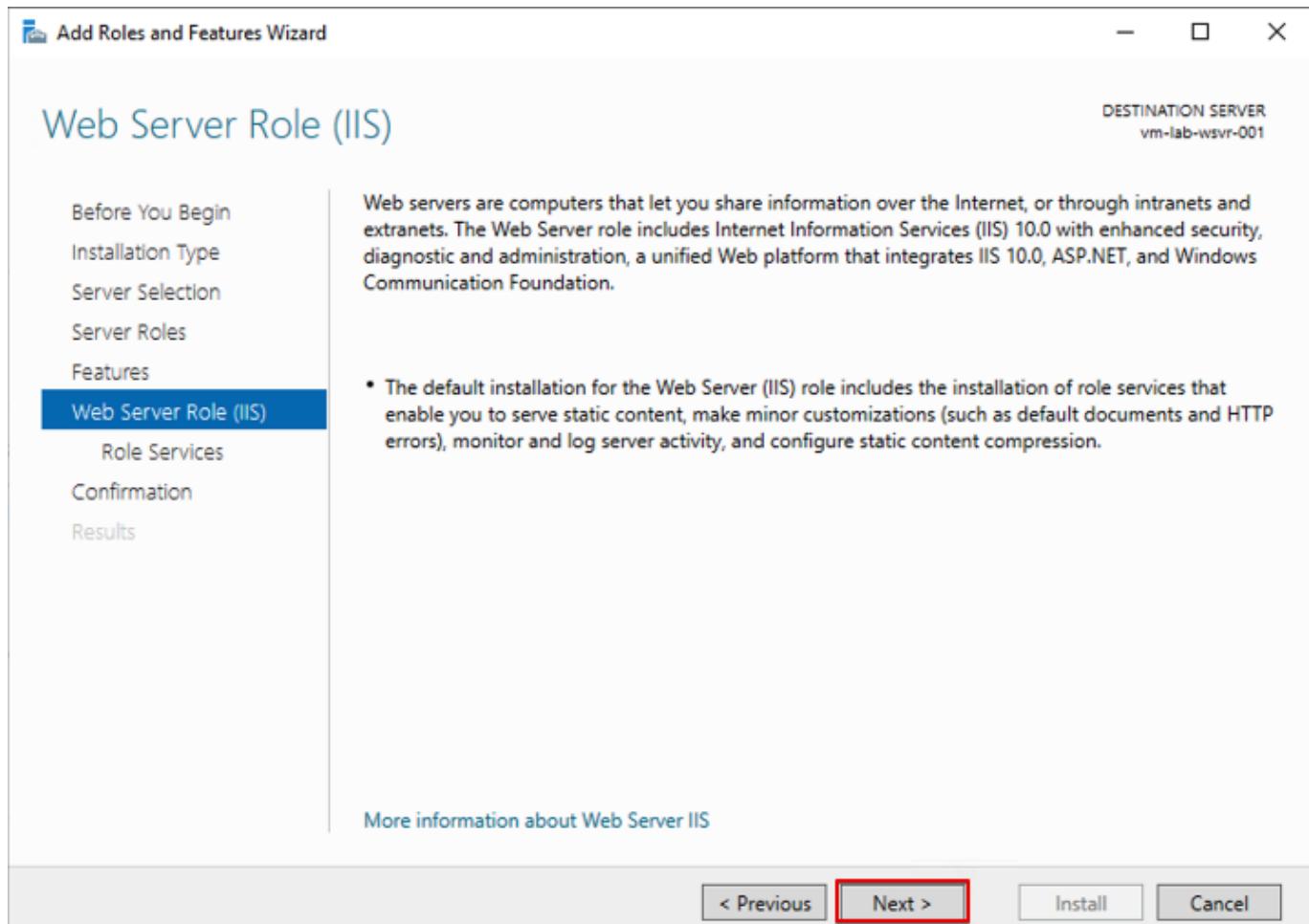
以下のとおり、 チェック済みとなればOKですので**Next**で進みます。



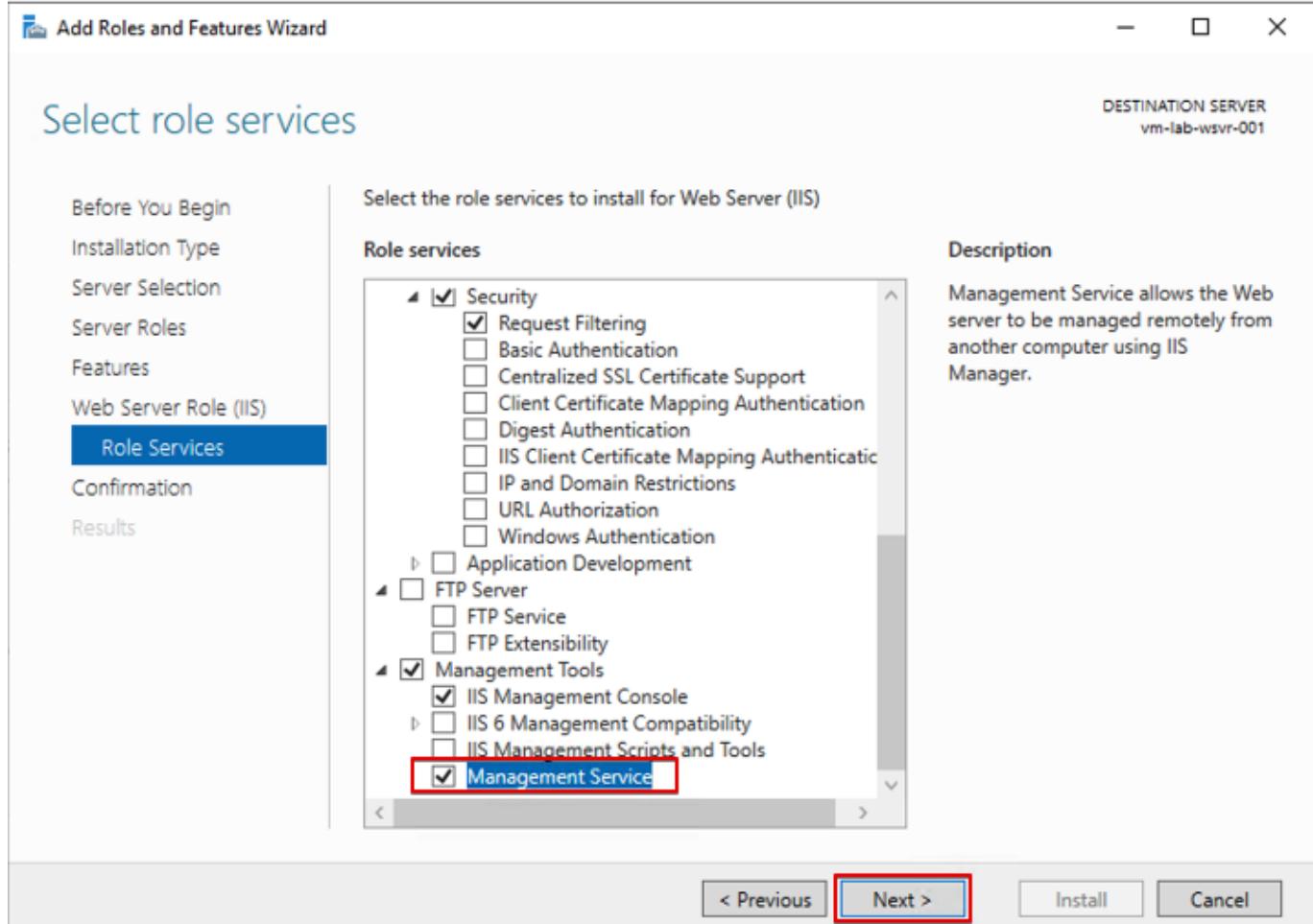
Featuresは変更しないので、Nextで進みます。



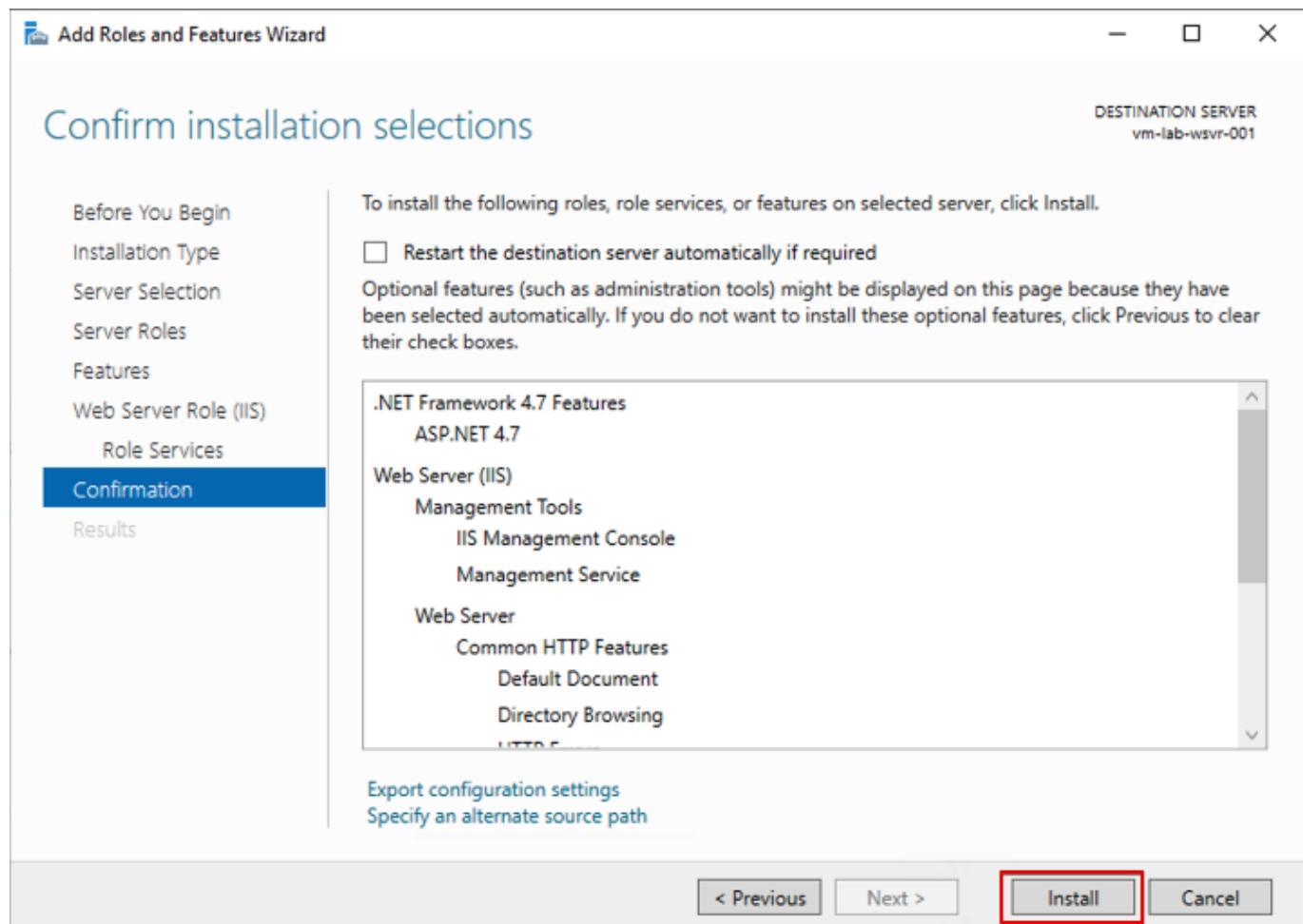
Web Server Role(IIS)もNextで進みます。



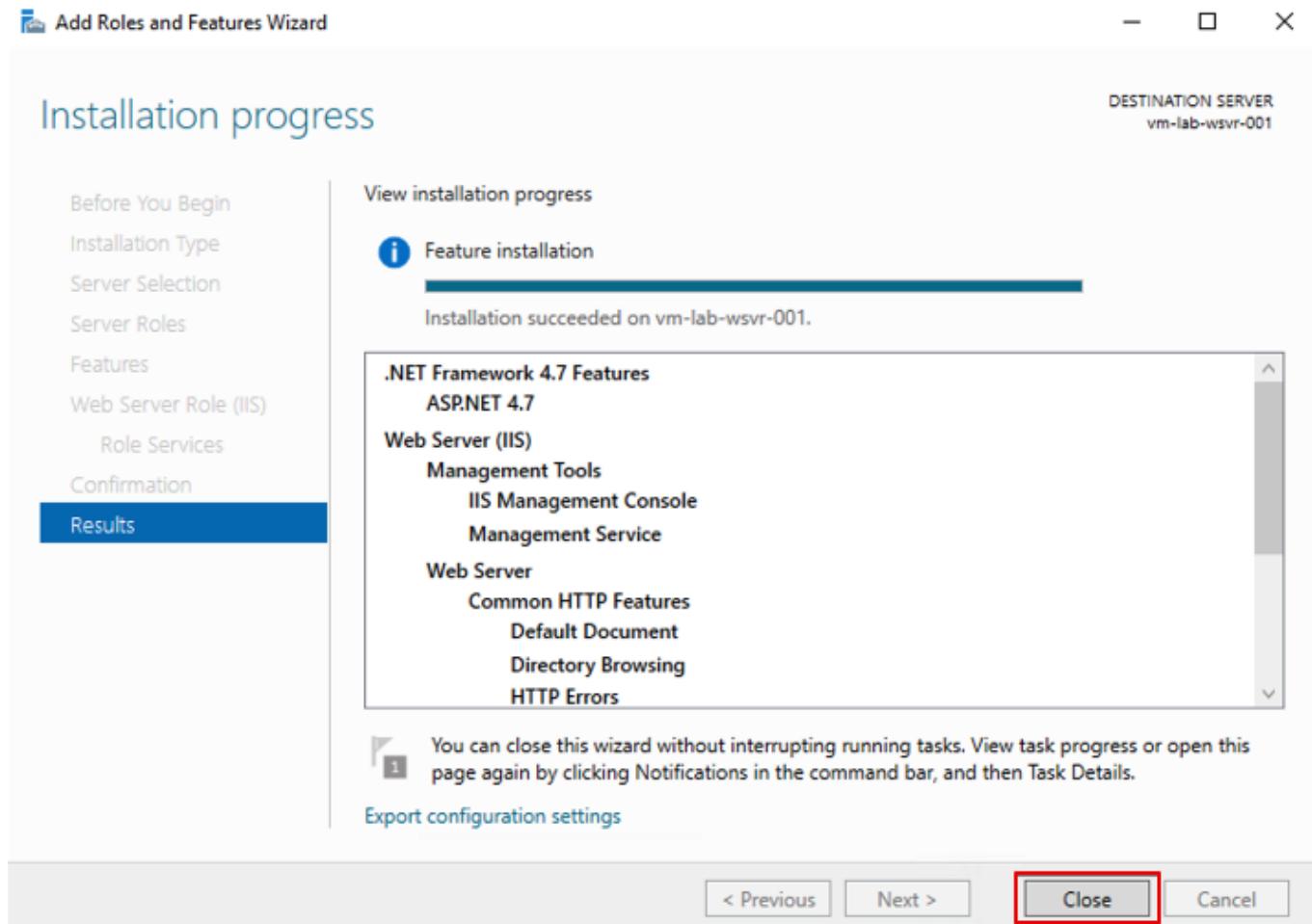
Role servicesではManagement Serviceを選択しチェック済みとなったらNextで進みます。



Installをクリックします。



Installation successとなればインストール完了なので**Close**で閉じます。



Azure Portalに戻り、動作しているか確認します。
検索したいサービスを入力し、Virtual Machinesを選択します。

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with navigation links like '概要', 'はじめに', '社内サポート', '個人用ラボ', '自分の仮想マシン', '要求可能な仮想マシン', 'すべての仮想マシン', 'セキュリティ通知', '個人用データ ディスク', '数式(再利用可能なベース)', '自分のシークレット', and '個人データ'. The main area is a search results page with a search bar at the top containing 'virtual'. The results are categorized under 'サービス' (Services). The 'Virtual Machines' option is highlighted with a red box. Other visible items include 'Virtual Machine Scale Sets', 'Virtual Network', 'eCare21 Virtual Care Platform', 'Virtual WAN', and 'Virtual machine scale set'. To the right, there are sections for 'Marketplace' (listing 'Virtual Network', 'eCare21 Virtual Care Platform', 'Virtual WAN', and 'Virtual machine scale set') and 'ドキュメント' (listing 'Windows Virtual Desktop に接続する Windows 10 または 7 ...', 'Azure のドキュメント | Microsoft Docs', 'Web クライアントを使用して Windows Virtual Desktop に接続 ...', and 'Azure 向けの Microsoft マルウェア対策 | Microsoft Docs'). At the bottom, there's a note: '該当結果が見つかりませんでした。' (No results found.)

対象のVMを選択し、VMの概要ページへ移動します。

新しい仮想マシンのリソース ブラウザーをお試しください。このエクスペリエンスは高速で、並べ替えとフィルターハンドルが強化されています。新しいエクスペリエンスを試すには、ここをクリックしてください。新しいエクスペリエンスでは、従来の仮想マシンが表示されないことにご注意ください。

サブスクリプション: tanabebe-dev-001

名前でフィルター...	すべてのリソース グループ	すべての種類	すべての場所	すべてのタグ	グループ化なし		
<input type="checkbox"/> 名前 ↑↓	<input type="checkbox"/> 種類 ↑↓	<input type="checkbox"/> 状態	<input type="checkbox"/> リソース グループ ↑↓	<input type="checkbox"/> 場所 ↑↓	<input type="checkbox"/> ソース	<input type="checkbox"/> メンテナンスの状態	<input type="checkbox"/> サブスクリプション ↑↓
<input type="checkbox"/> vm-lab-wsvr-001	仮想マシン	実行中	devtest-labs-env-001-vm-lab-ws...	東日本	Marketplace	-	tanabebe-dev-001
<input type="checkbox"/> vm-win-core-001	仮想マシン	停止済み(割り当て解除)	devtestlabos-001-vm-win-core-...	東日本	Marketplace	-	tanabebe-dev-001
<input type="checkbox"/> vm-winserve-001	仮想マシン	停止済み(割り当て解除)	devtestlabos-001-vm-winserve-...	東日本	Marketplace	-	tanabebe-dev-001
<input type="checkbox"/> vm-winsv-blazor-001	仮想マシン	停止済み(割り当て解除)	vm-winsv-blazor-001_group	東日本	Marketplace	-	tanabebe-dev-001

パブリックIPをクリップボードにコピーして、ブラウザの別タブに貼り付けます。

新しい仮想マシンのリソース ブラウザーをお試しください。このエクスペリエンスは高速で、並べ替えとフィルターハンドルが強化されています。新しいエクスペリエンスを試すには、ここをクリックしてください。新しいエクスペリエンスでは、従来の仮想マシンが表示されないことにご注意ください。

vm-lab-wsvr-001

概要

vm-lab-wsvr-001

接続

リソース グループ (変更) : devtest-labs-env-001-vm-lab-ws-001-192844
状態 : 実行中
場所 : 東日本
サブスクリプション (変更) : tanabebe-dev-001
サブスクリプション ID :
タグ (変更) : タグを追加するにはここをクリック

設定

ネットワーク

接続

ディスク

サイズ

セキュリティ

Advisor の推奨事項

拡張機能

継続的デリバリー

可用性とスケーリング

構成

ID

プロパティ

ロック

プロパティ

監視

機能

推薦事項 (3 個)

チュートリアル

仮想マシン

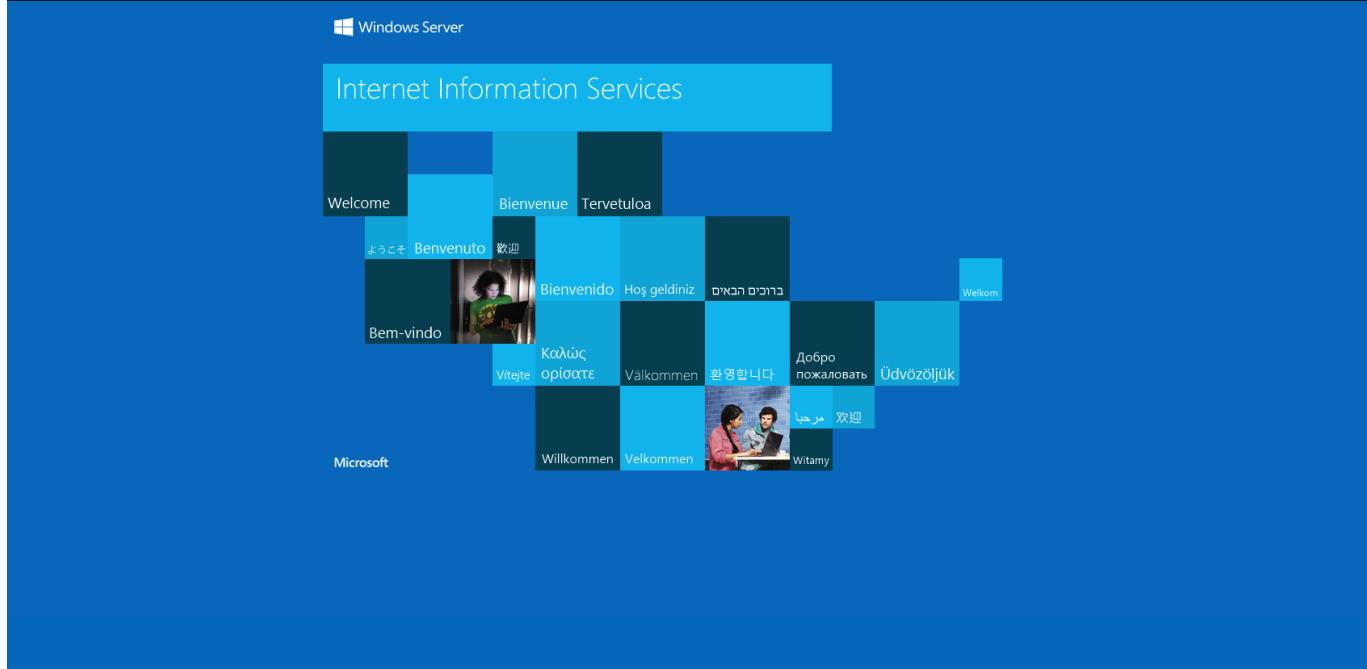
コンピューター名: vm-lab-ws-001
オペレーティング システム: Windows (Windows Server 2019 Datacenter)
SKU: 2019-Datacenter
パブリッシャー: MicrosoftWindowsServer
VM の生成: V1
エージェントの状態: Ready
エージェントのバージョン: 2.7.41491.993
ホスト: なし
近接配置グループ: N/A

ネットワーク

パブリック IP アドレス: 10.0.0.4
パブリック IP アドレス (IPv6): -
プライベート IP アドレス: 10.0.0.4
プライベート IP アドレス (IPv6): -
仮想ネットワーク/サブネット: D1/devtest-labs-env-001/D1/devtest-labs-env-001
DNS 名: 構成

以下のように画面が表示されればOKです。

次は再度VMに接続し、サーバーホスティング用の.NET Coreをインストールしていきます。



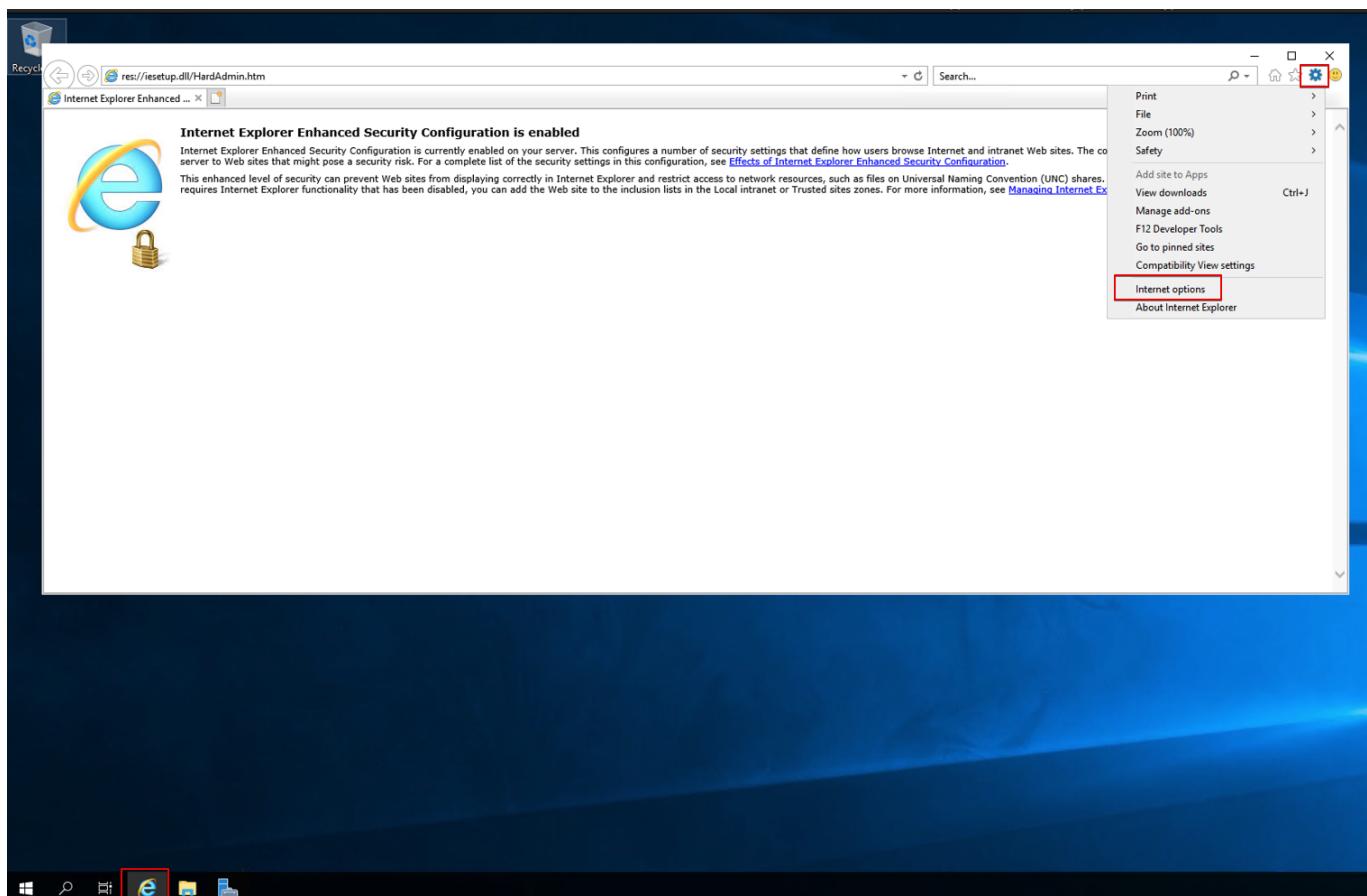
3.5. .NET Coreのインストール

再びRDPでWindows Serverを操作していきます。

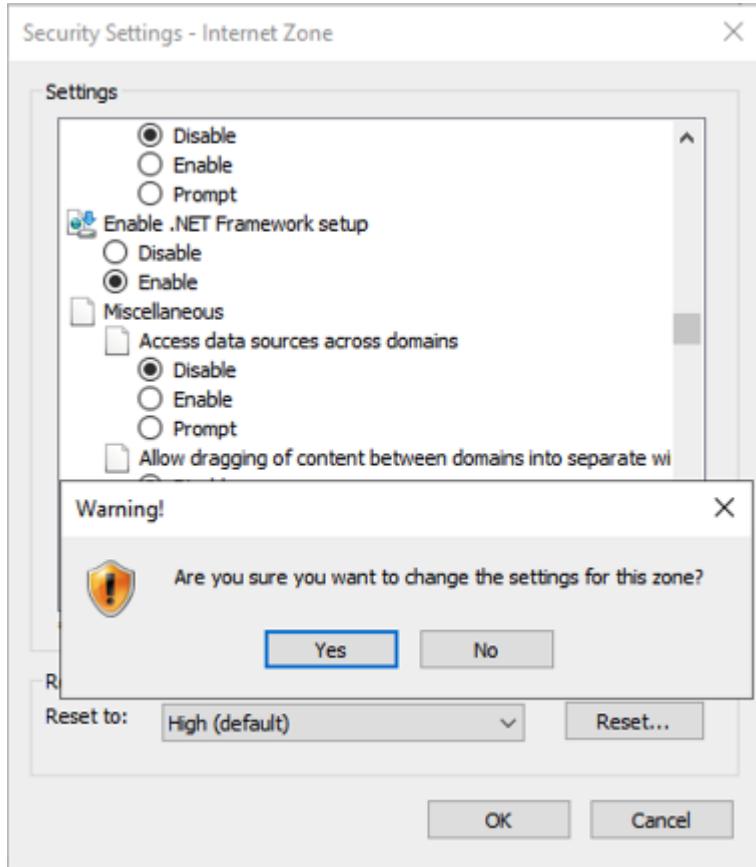
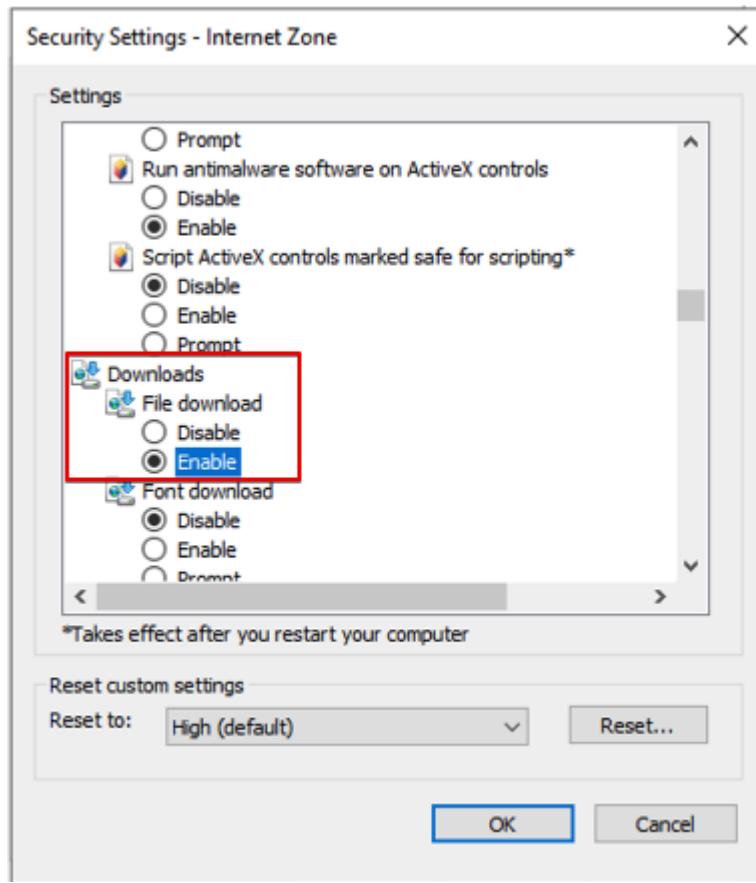
今回はBlazor ServerのアプリをIISに実行してもらう必要があるため、.NET Coreが必要となります。

IEを開き、Internet optionsを選択します。

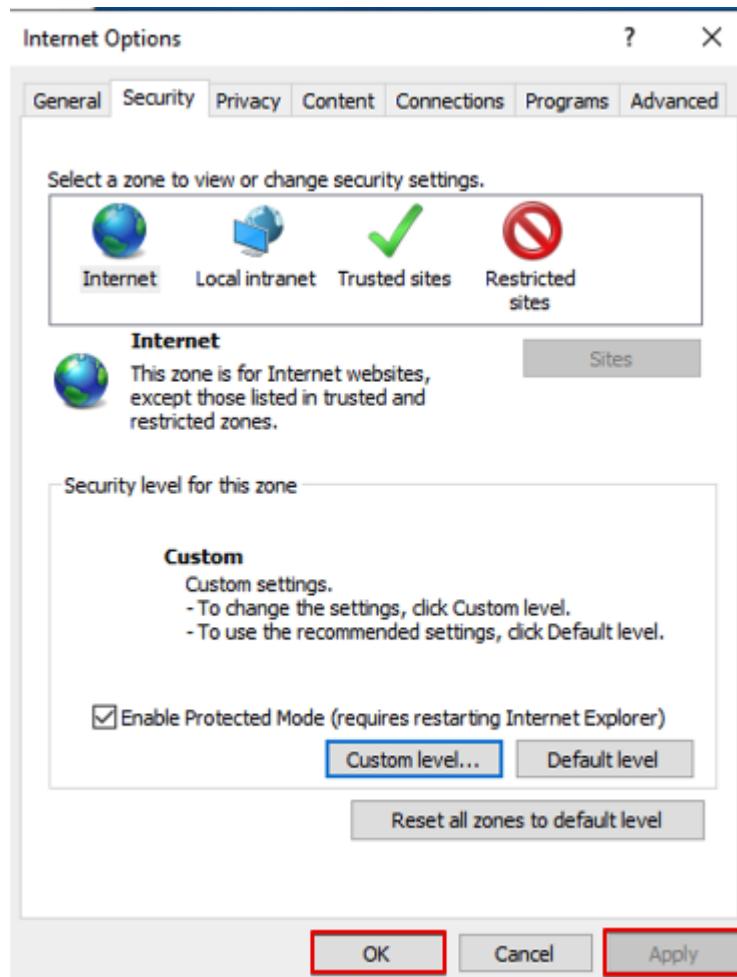
インターネットからファイルのダウンロードを行うので、IEの設定を変更します。



File downloadのチェックボックスをEnableにします。警告が出ますが、Yesを押して下さい。



Applyをクリックした後、OKで閉じて下さい。



IEにこちらのURLをコピペして下さい。.NET CoreのRuntimeが案内されている画面になるので、Windowsの**Hosting Bundle**をクリックします。

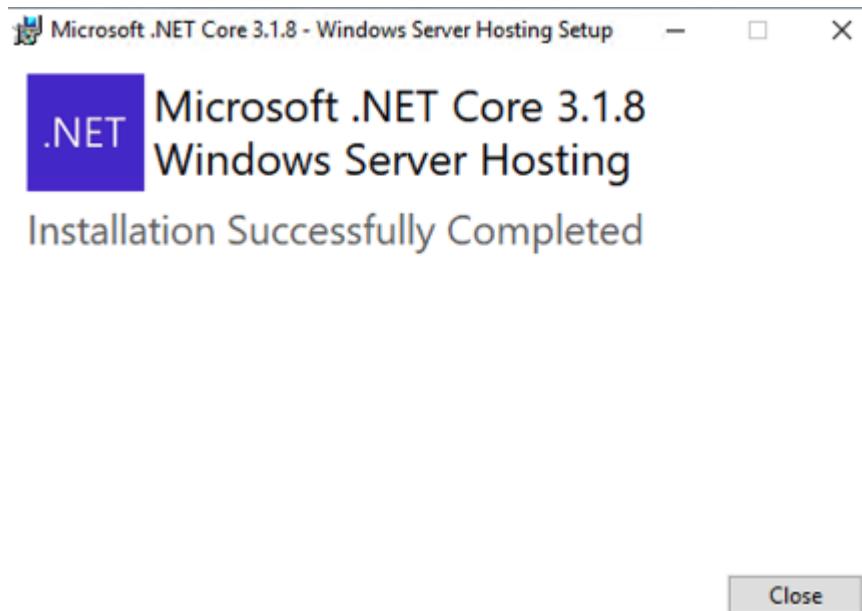
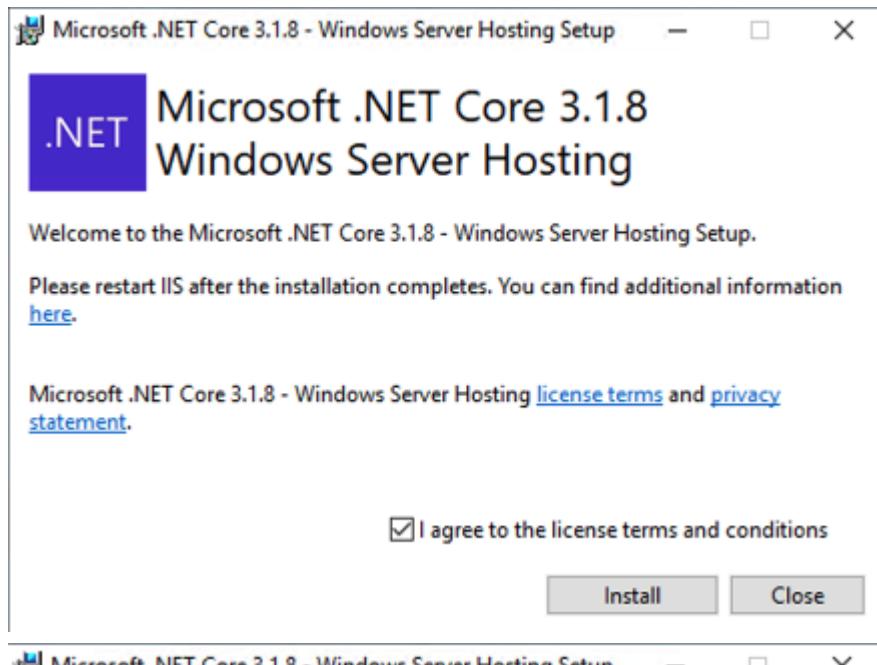
The screenshot shows the Microsoft .NET Core download page at <https://dotnet.microsoft.com/download/dotnet-core/3.1>. The page displays release information for v3.1.8, including a 'Security patch' link and release notes. It lists 'Included in' Visual Studio 16.7.3, 'Included runtimes' (.NET Core Runtime 3.1.8, ASP.NET Core Runtime 3.1.8, Desktop Runtime 3.1.8), and 'Language support' (C# 8.0, F# 4.7). It also provides 'Build apps - SDK' and 'Run apps - Runtime' sections. The 'Run apps - Runtime' section for Windows highlights the 'Hosting Bundle' link, which is also highlighted with a red box. The 'Hosting Bundle' link is located under the 'OS' column for Windows, next to 'x64' and 'x86'.

リンクをクリックして、ファイルをダウンロードします。（ファイルの保存先は任意）

The screenshot shows a web browser displaying the Microsoft .NET Core download page. The URL in the address bar is <https://dotnet.microsoft.com/download/dotnet-core/thank-you/runtime-aspnetcore-3.1.8-windows-hosting-bundle-installer>. The page title is "Download ASP.NET Core 3.x". The main content features a purple background with white text that reads "Thanks for downloading ASP.NET Core 3.1 Runtime (v3.1.8) - Windows Hosting Bundle Installer!". Below this, a note states: "ASP.NET Core Hosting Bundle contains everything you need to run existing web/server apps. The bundle includes the .NET Core runtime, the ASP.NET Core runtime, and if installed on a machine with IIS it will also add the [ASP.NET Core IIS Module](#). If you only want the .NET Core or ASP.NET Core runtime, you'll find them in [.NET Core 3.1 Downloads](#)". A "Direct link" button is highlighted with a red box, showing the URL <https://download.visualstudio.microsoft.com/download/pr/854cbd11-4b96-4a44-9664-b95991c0c4f7/8ec4944a5bd770faba2f769e647b1e6e/dotnet-hosting-3.1.8-win.exe>. To the right of the link are "Copy" and "Checksum (SHA512)" buttons. Below the link are five small colored icons representing different file types.

ダウンロードした **dotnet-hosting-3.1.8-win.exe** をダブルクリックで実行します。

同意のチェックを入れ、インストールします。完了したら閉じましょう。

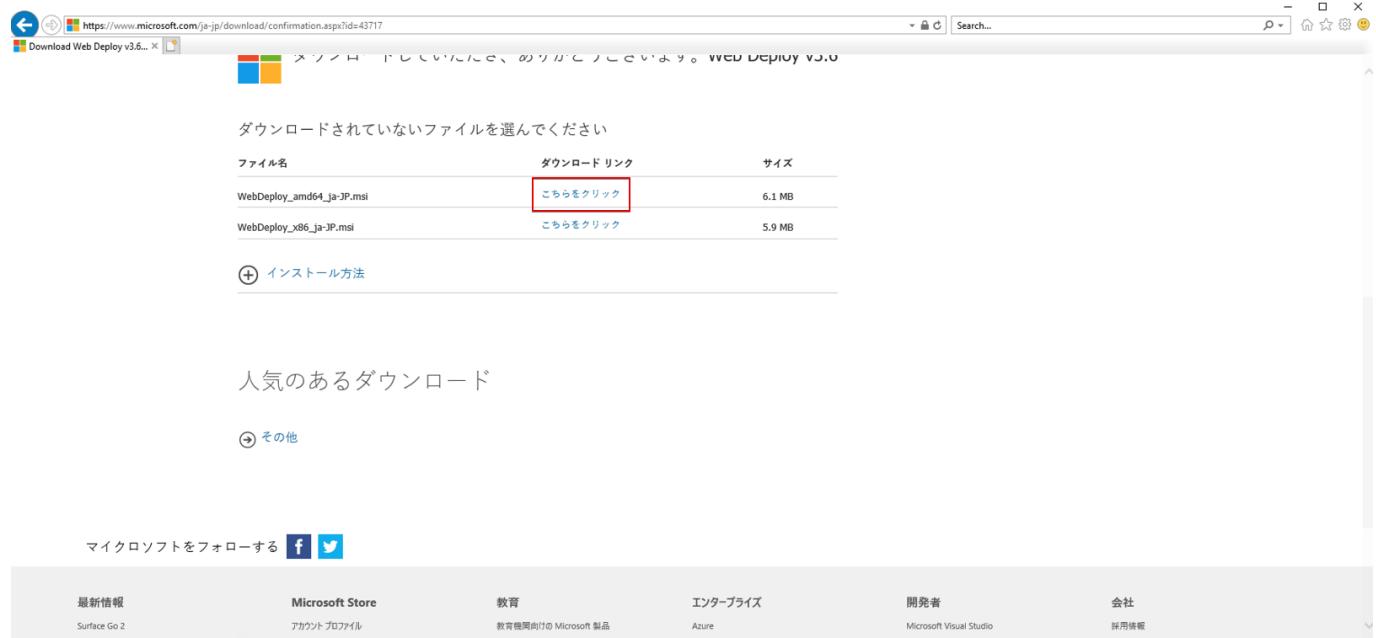


3.6. Web Deployのインストール

今度はVisual StudioからAzureにデプロイを簡略化するためにWeb Deployをインストールします。

[こちらのURL](#)をWindowsServerのIEへコピペして移動します。

セキュリティ面での制限で画面が崩れている場合がありますが、ダウンロード出来るので以下の64bit側をダウンロードします。



ダウンロードされていないファイルを選んでください

ファイル名	ダウンロード リンク	サイズ
WebDeploy_amd64_ja-JP.msi	こちらをクリック	6.1 MB
WebDeploy_x86_ja-JP.msi	こちらをクリック	5.9 MB

[+ インストール方法](#)

人気のあるダウンロード

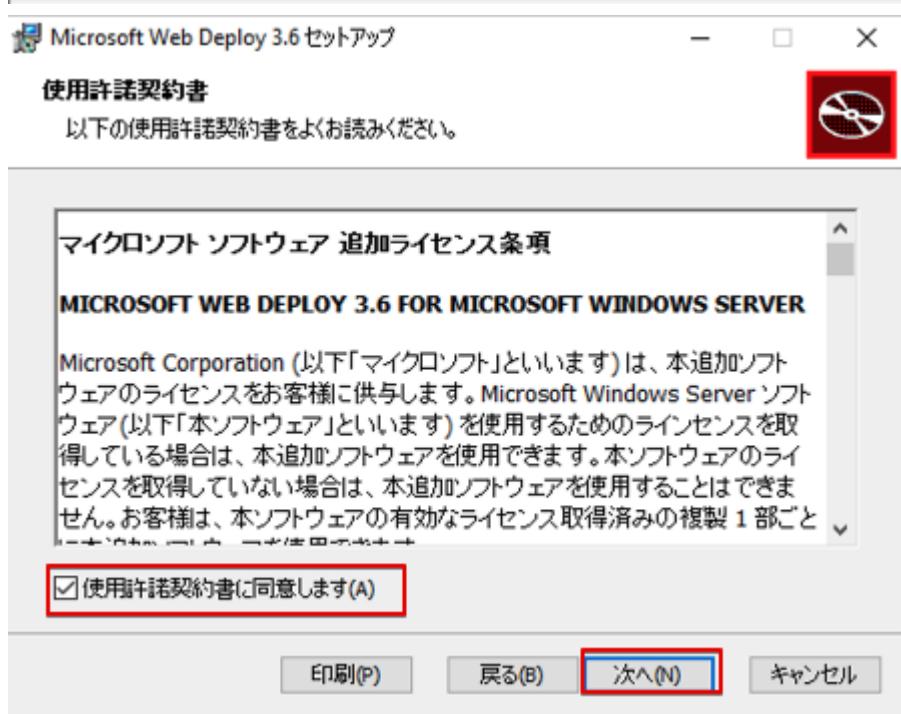
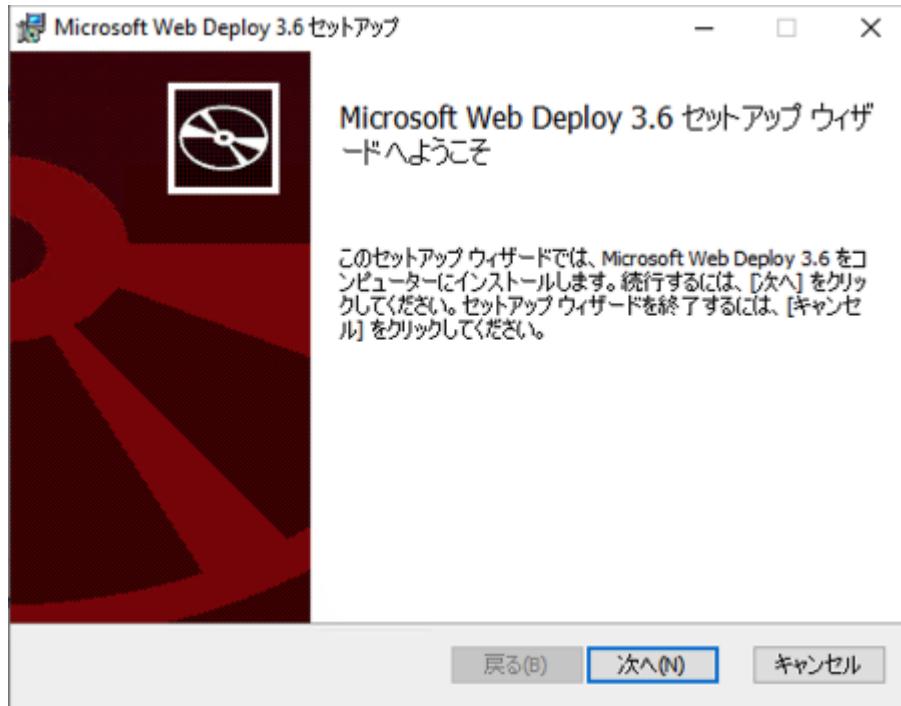
[その他](#)

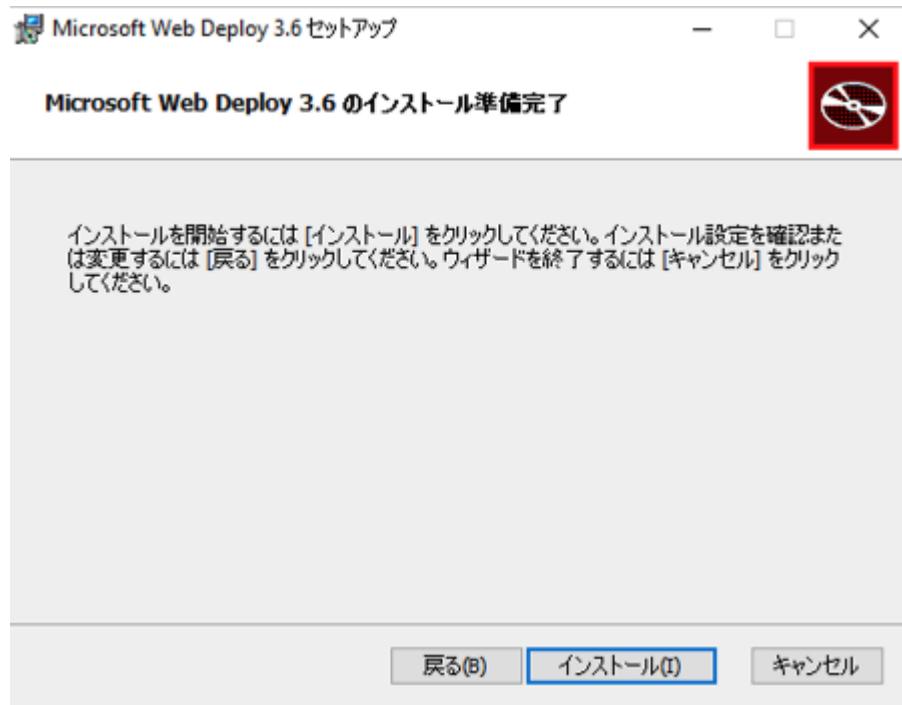
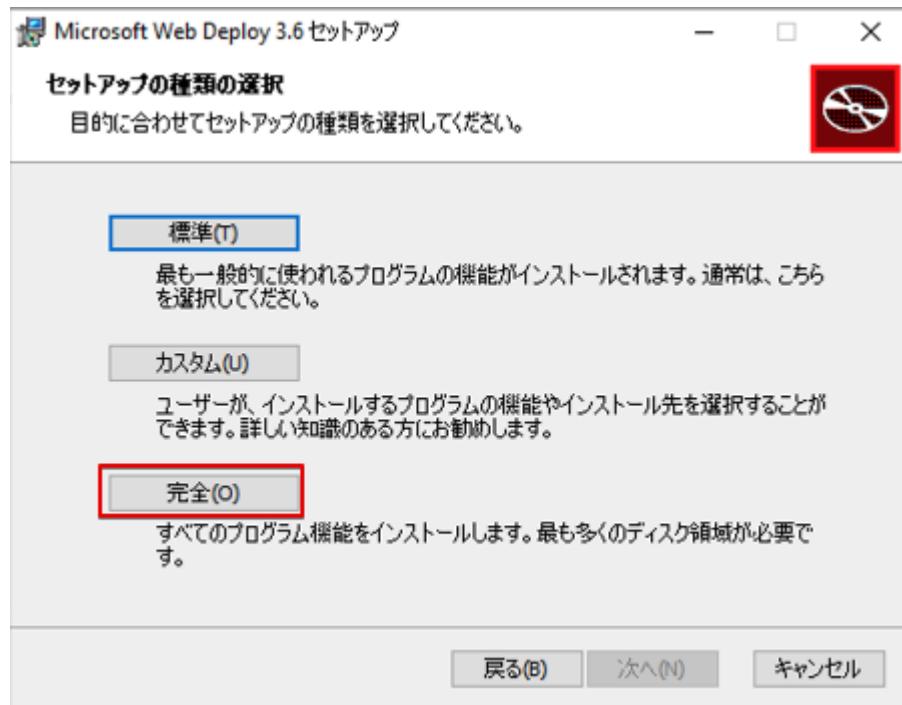
マイクロソフトをフォローする [f](#) [t](#)

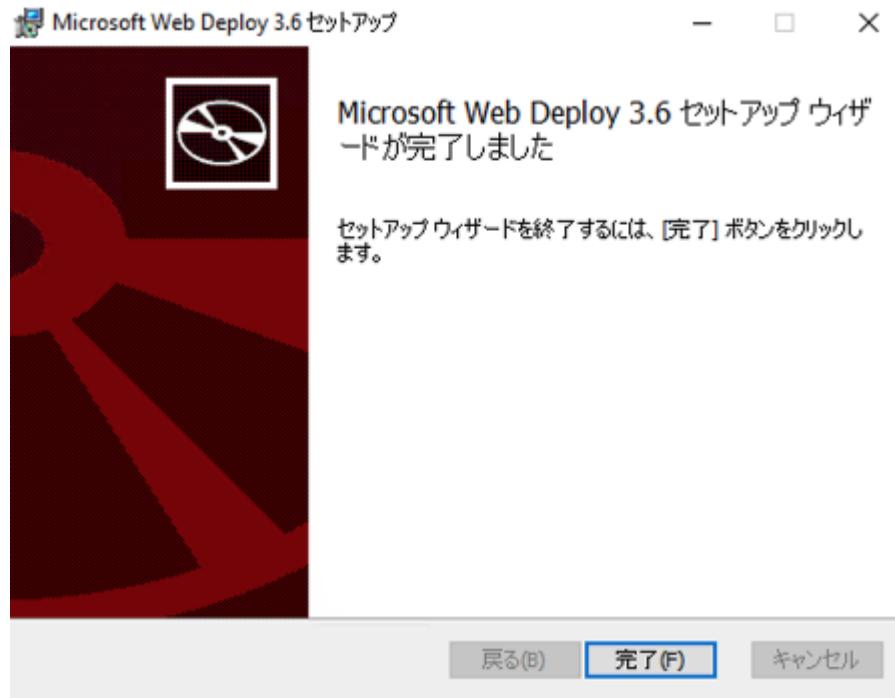
最新情報 Microsoft Store 教育 エンタープライズ 開発者 会社

Surface Go 2 アカウント プロファイル 教育機関向けの Microsoft 製品 Azure Microsoft Visual Studio 担当情報

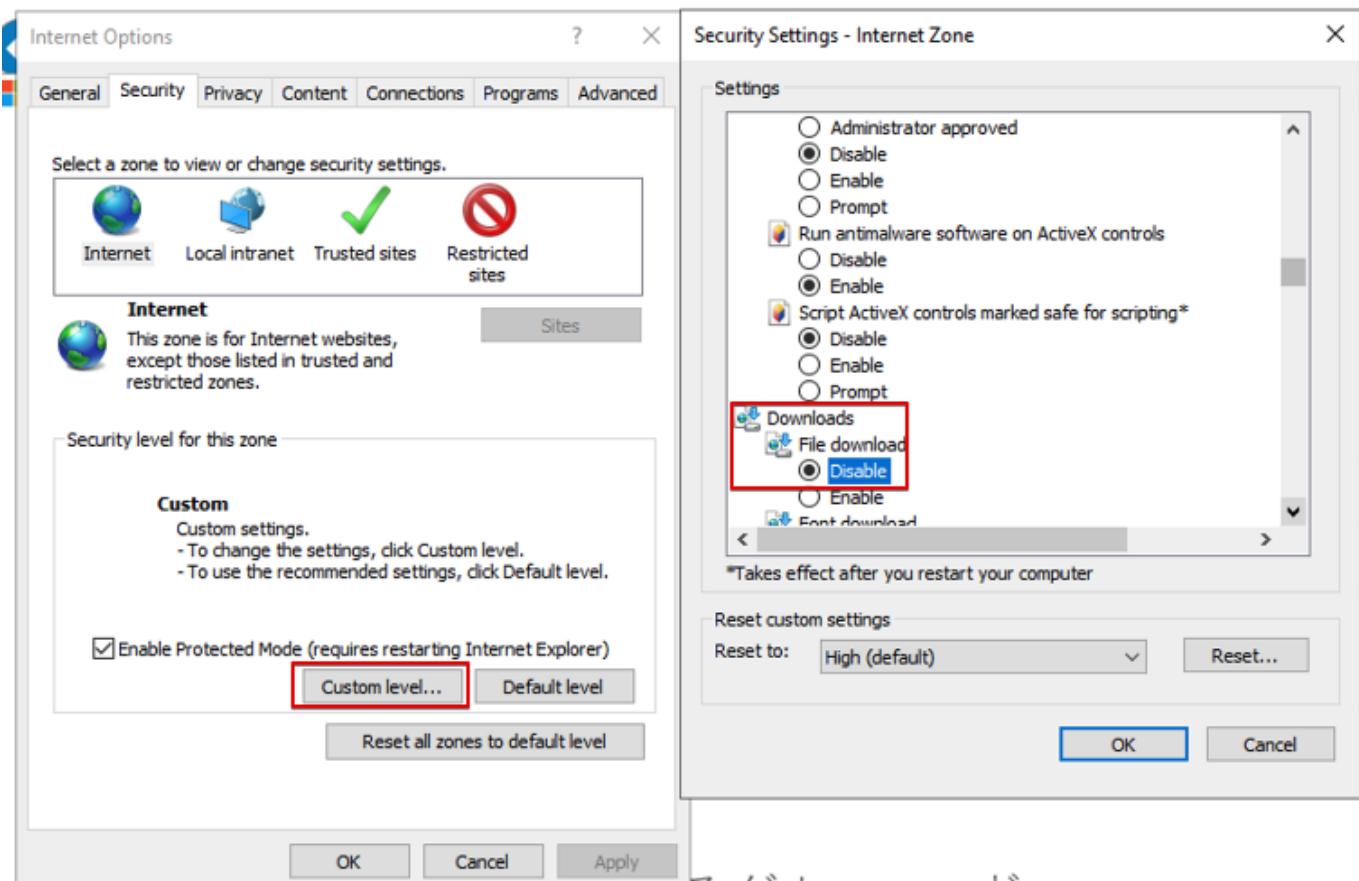
ダウンロードした [WebDeoloy_amd64_ja-JP.msi](#) をダブルクリックして以下の通り、インストールします。
※標準インストールではないことに注意してください。





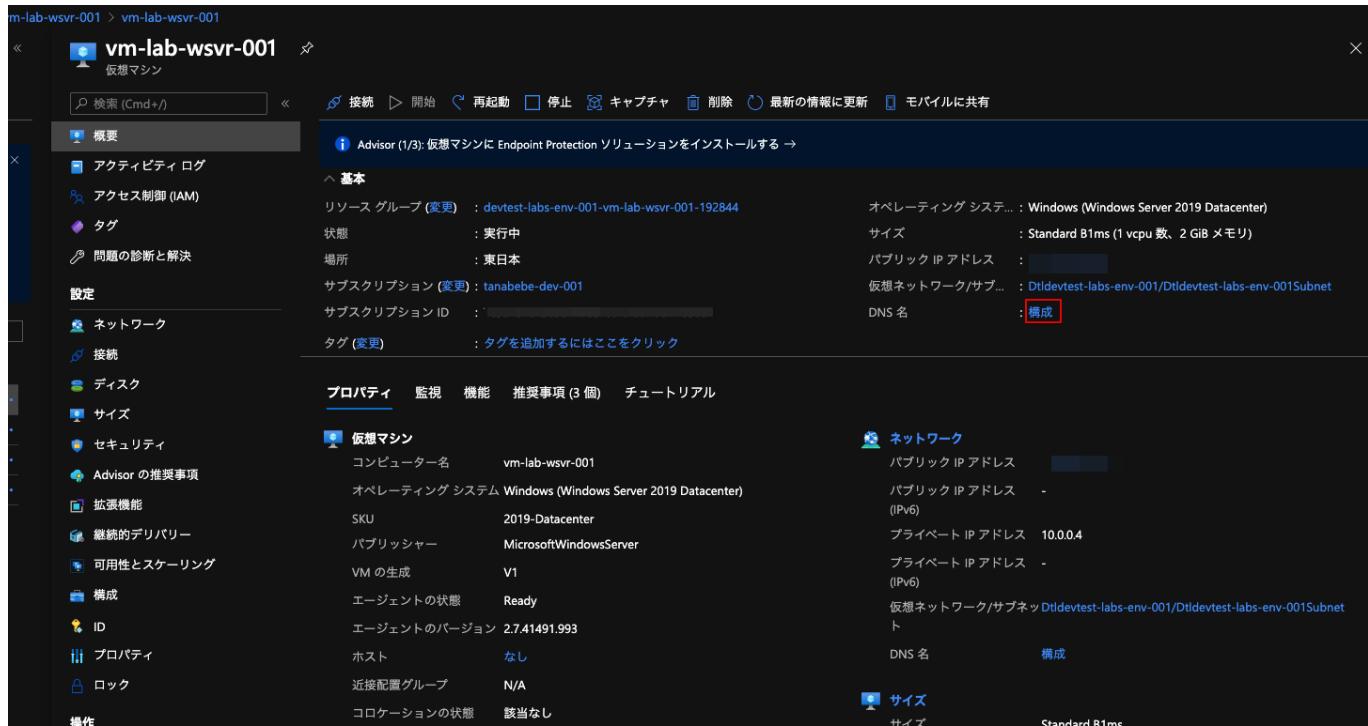


以降はファイルをダウンロードしないのでIEを開き、**Internet options**を戻しておきます。



3.7. VMのDNSを構成する

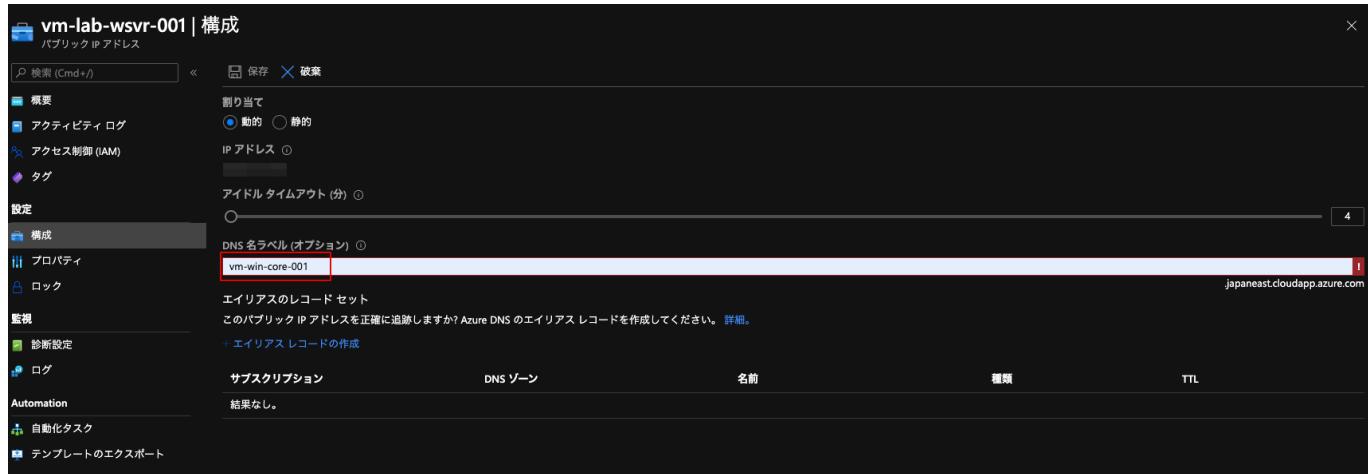
Azure PortalからVirtual Machinesで作成したVMを選択し、DNS名の構成をクリック。



Visual Studioから発行するときにDNSでアクセス出来ないと駄目なので、任意のDNS名を付けましょう。

作成したVMのマシン名がわかりやすくて良いと思います。変更したら**保存**をクリックします。

割り当ては動的としていますが、VMが再起動してもDNSで紐付くので問題ないです。



3.8. Visual Studio で Blazor Server アプリを作成する

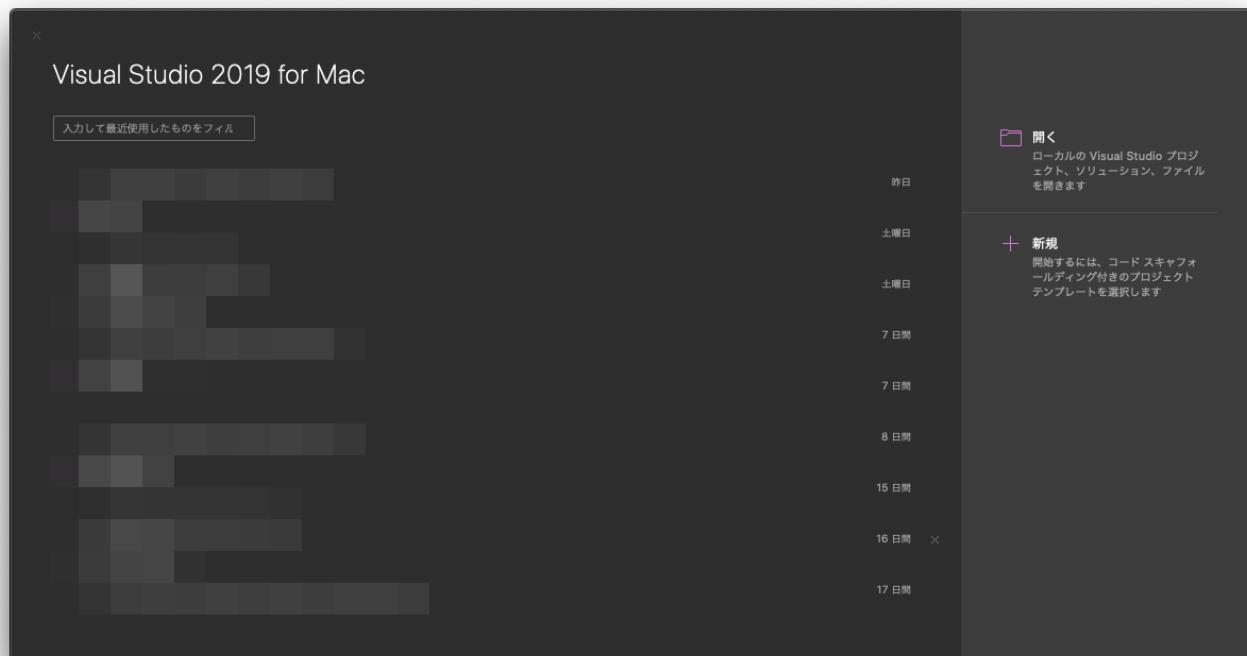
ここでWindowsとMacで手順が大きく変わりますので、自身の環境に合わせて読み進めてください。

Macを使用している方

Visual Studio for MacではAzure VMへWebデプロイの機能が追いついていないため、別の対応方法を記載しています。

Windowsは全てVisual Studioで完結しますが、Macは少し違う手法を取ります。

プロジェクトの新規作成を選択します。



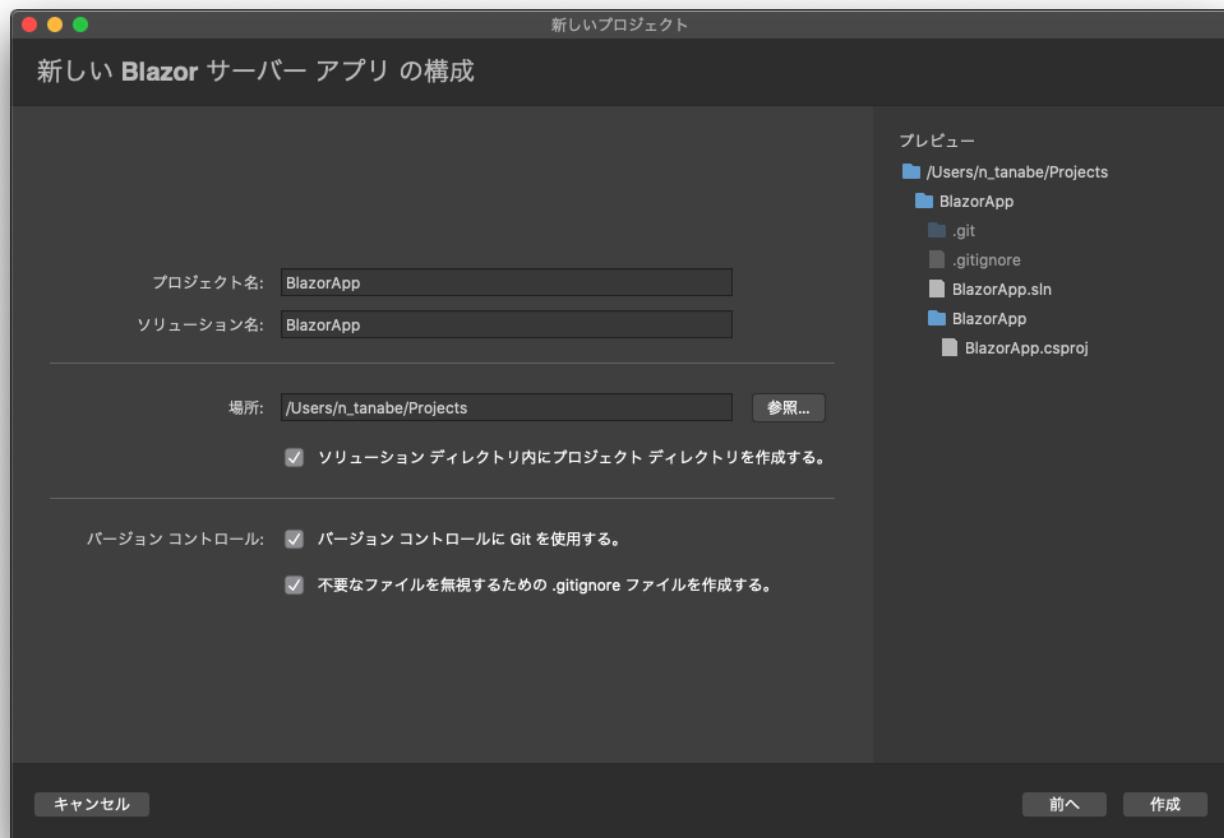
Blazor Serverを選択します。



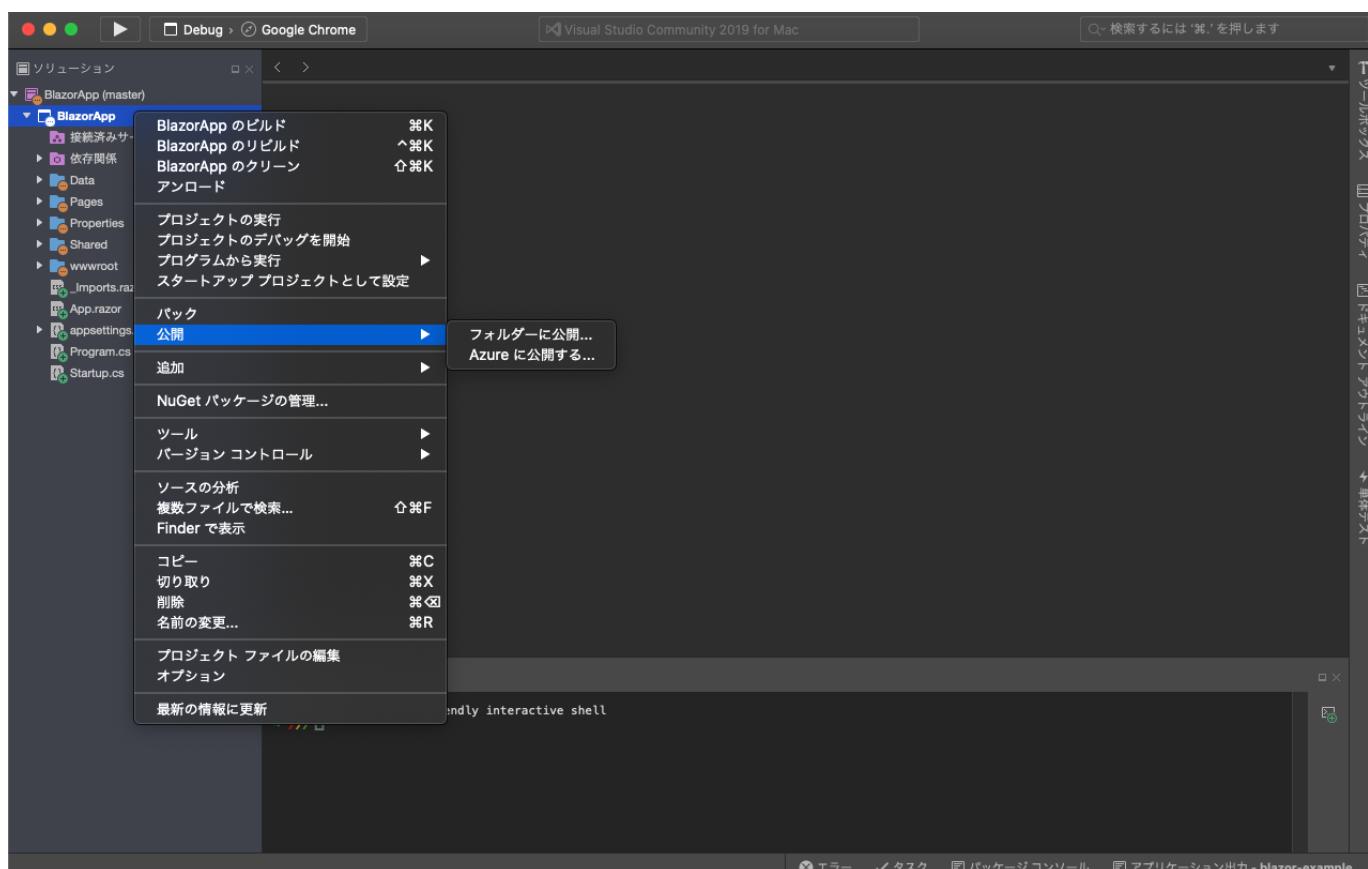
認証なしでOKです。



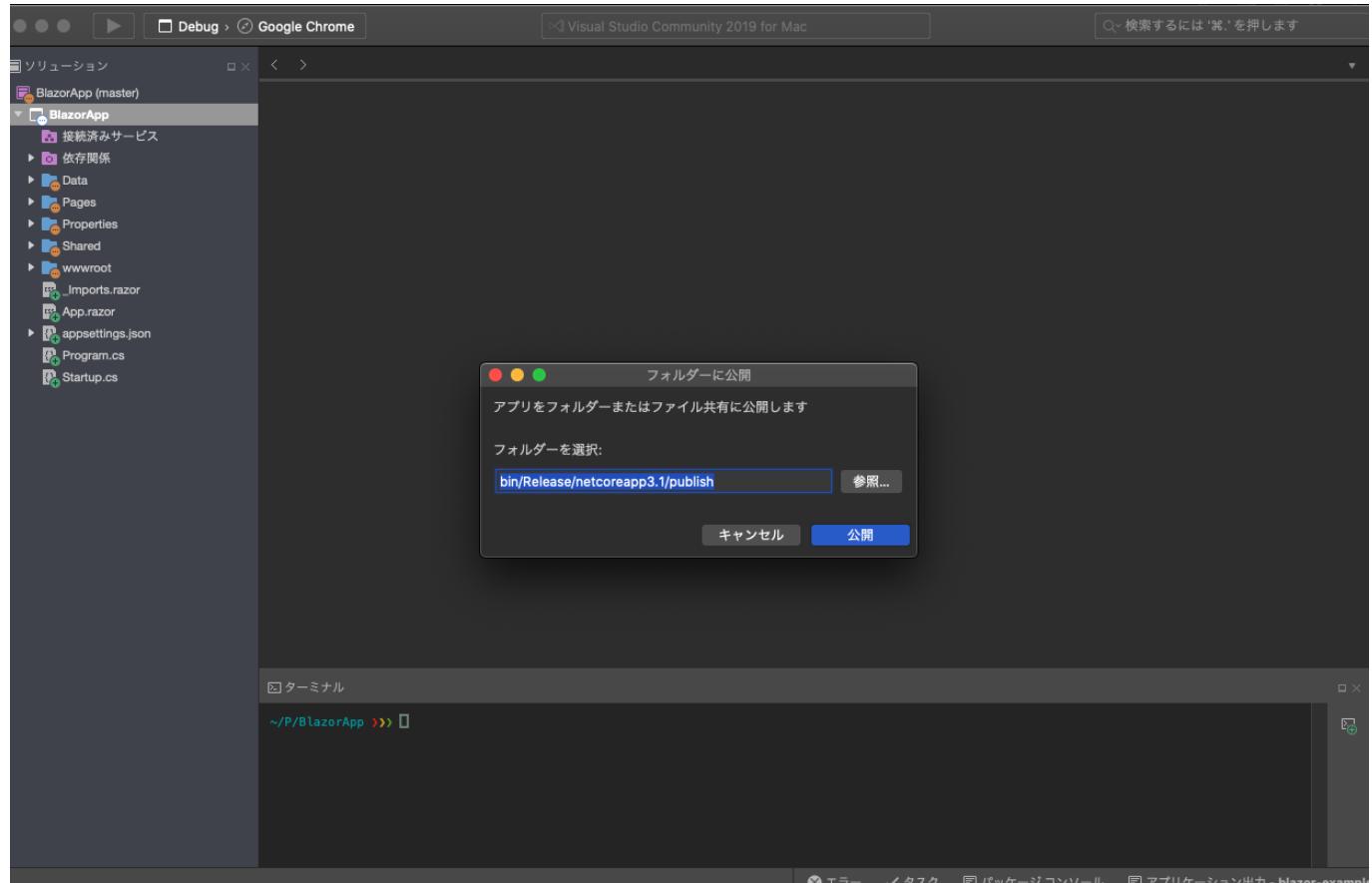
プロジェクト名や場所は任意なので、好きな名称を入力してください。



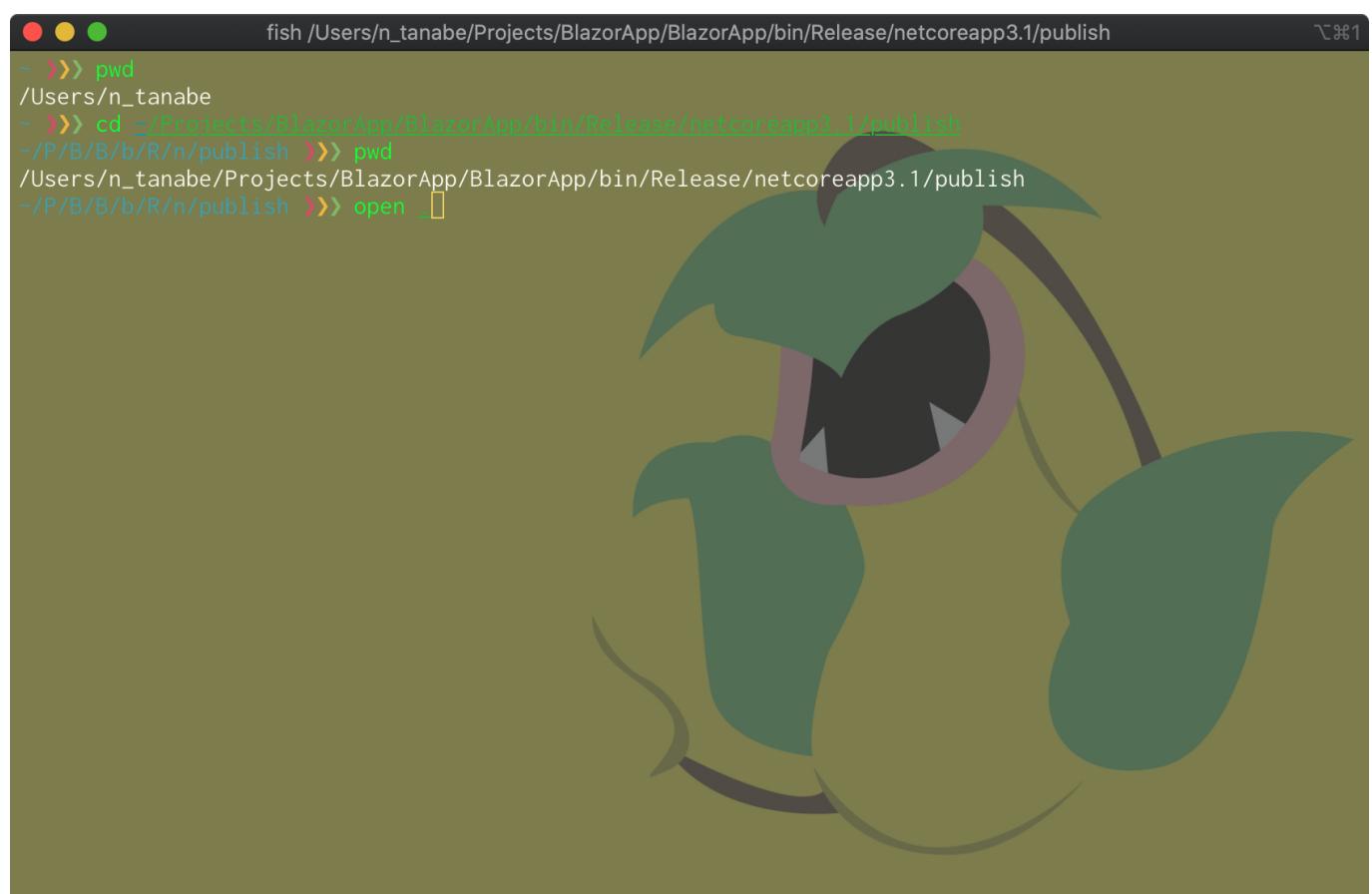
プロジェクトが作成されたらソリューションで右クリック->公開->フォルダに公開を選択してください。



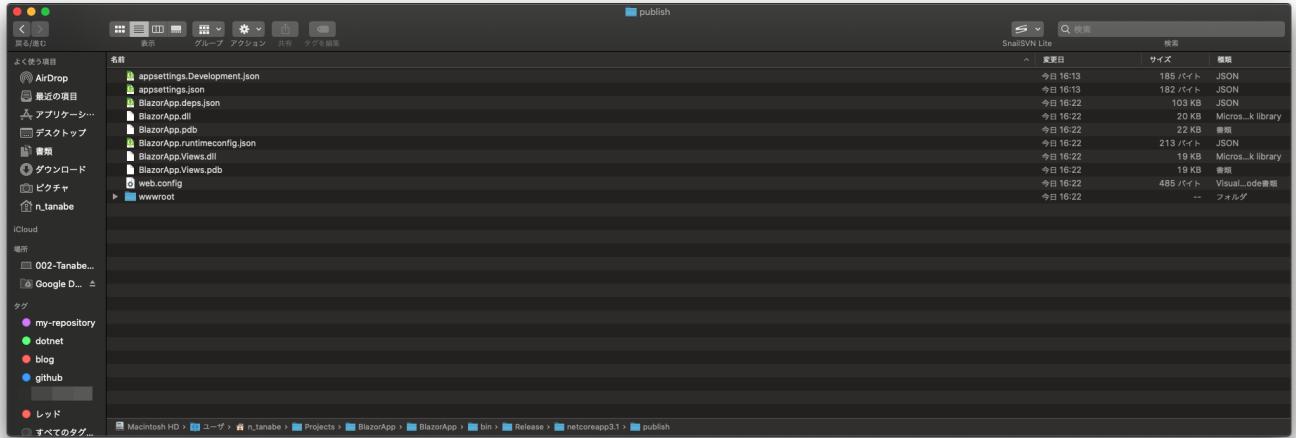
公開先のフォルダを選択します。
デフォルトのまま変更していないですが、任意です。



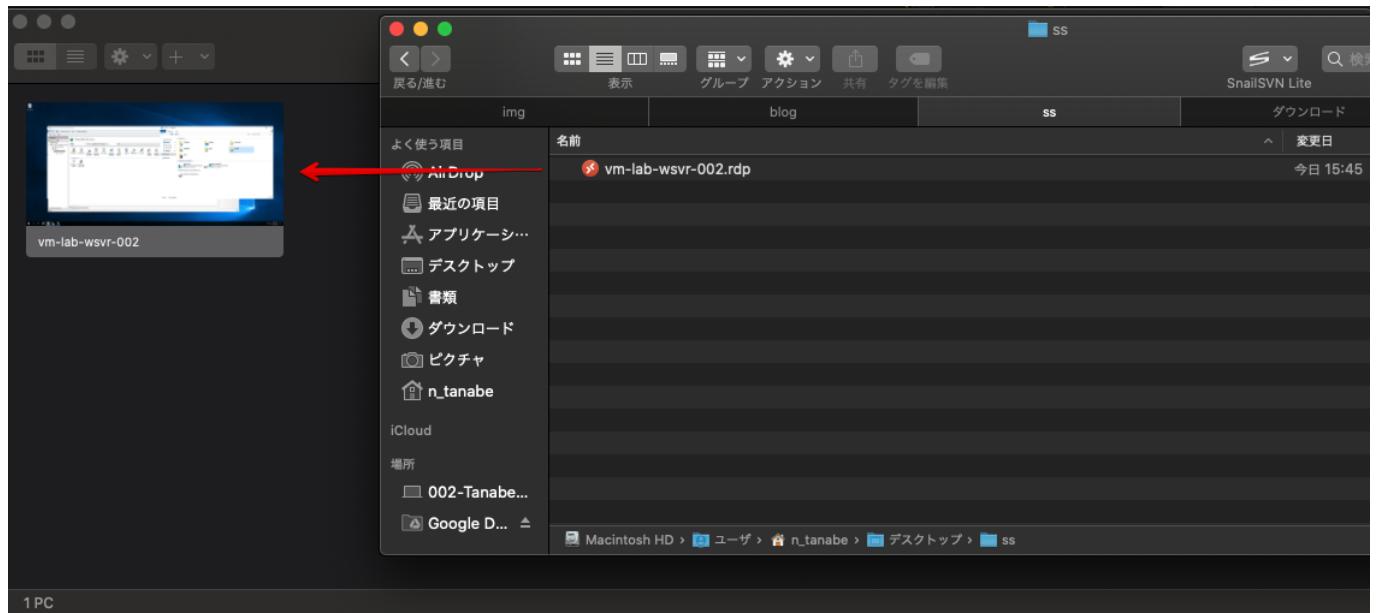
作成したプロジェクト内まで移動し、Finderを開きます。



以下のファイルをVM（Windows Server）へアタッチして、コピーしていきます。



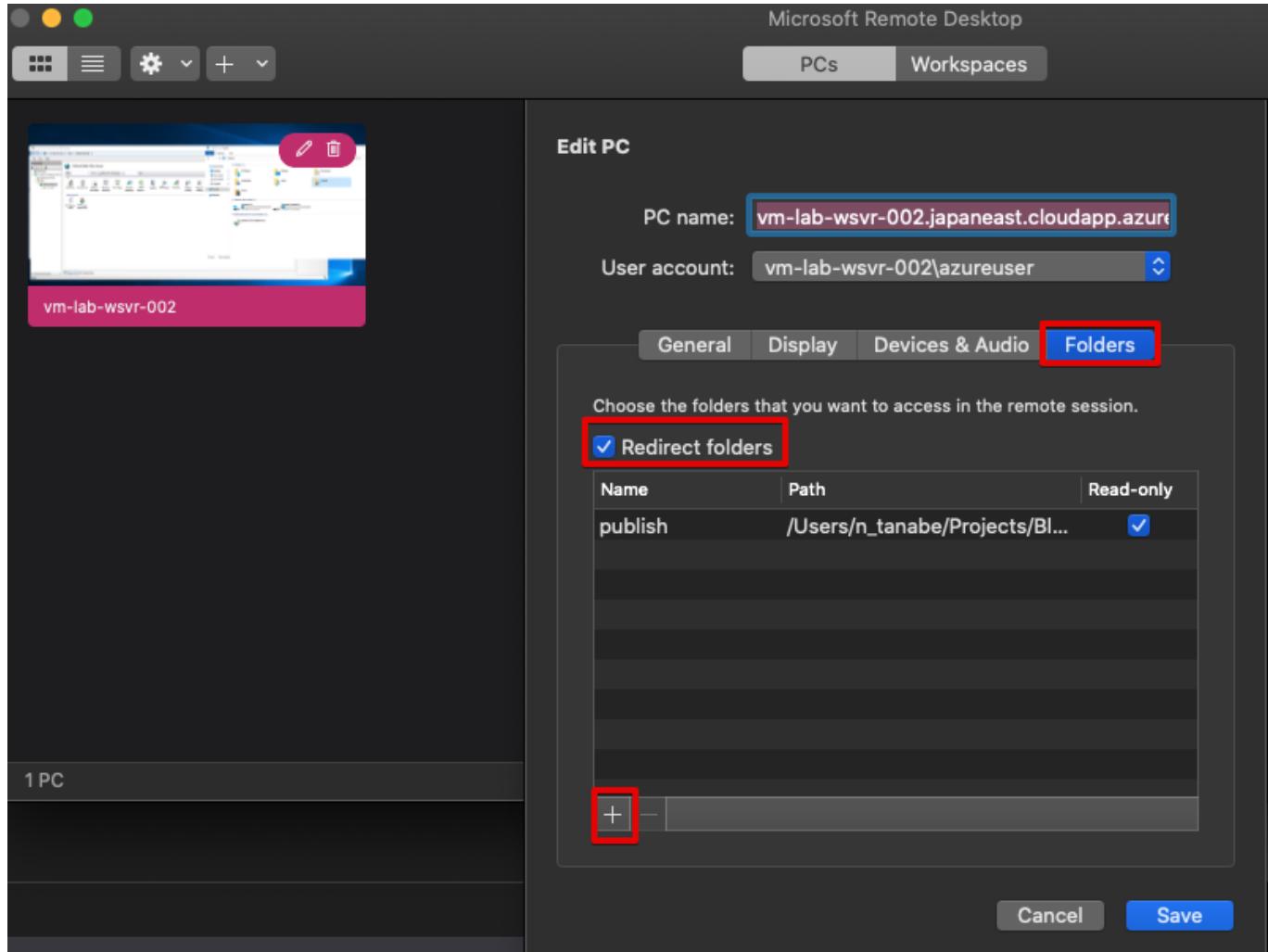
VMに接続するときに使用した.rdpファイルをMicrosoft Remote Desktopへドラッグ&ドロップします。



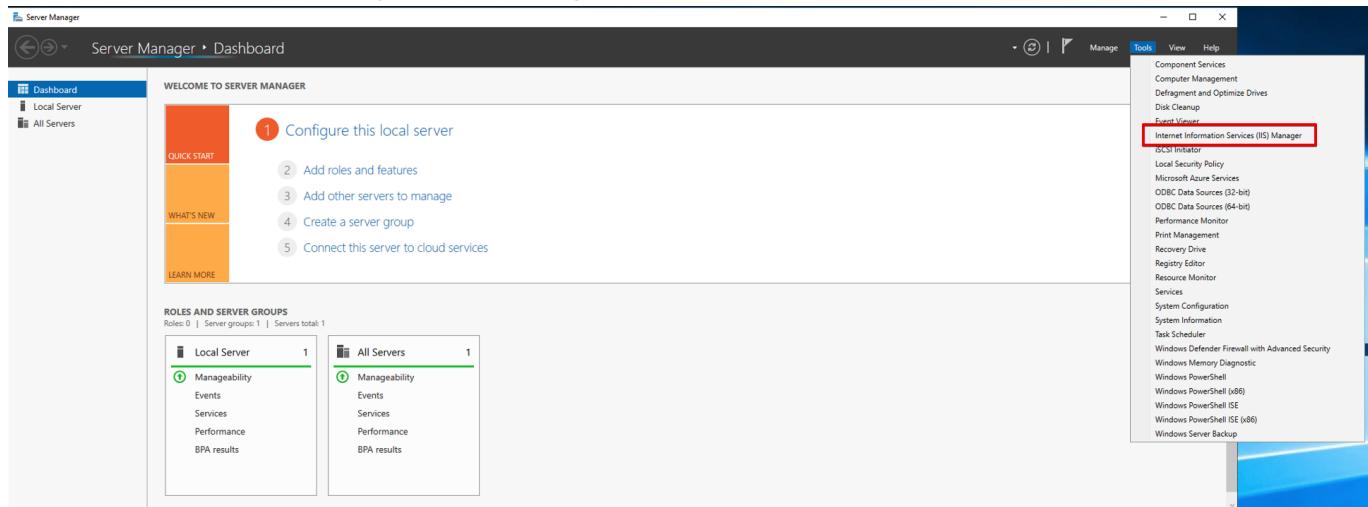
パネルを右クリックし、Editを選択、Foldersのタブを選択し、Redirect foldersにチェックを付けます。

[+]ボタンをクリックして、Visual Studioで公開先で選択したフォルダを指定し、VMから誤ってファイルが削除され

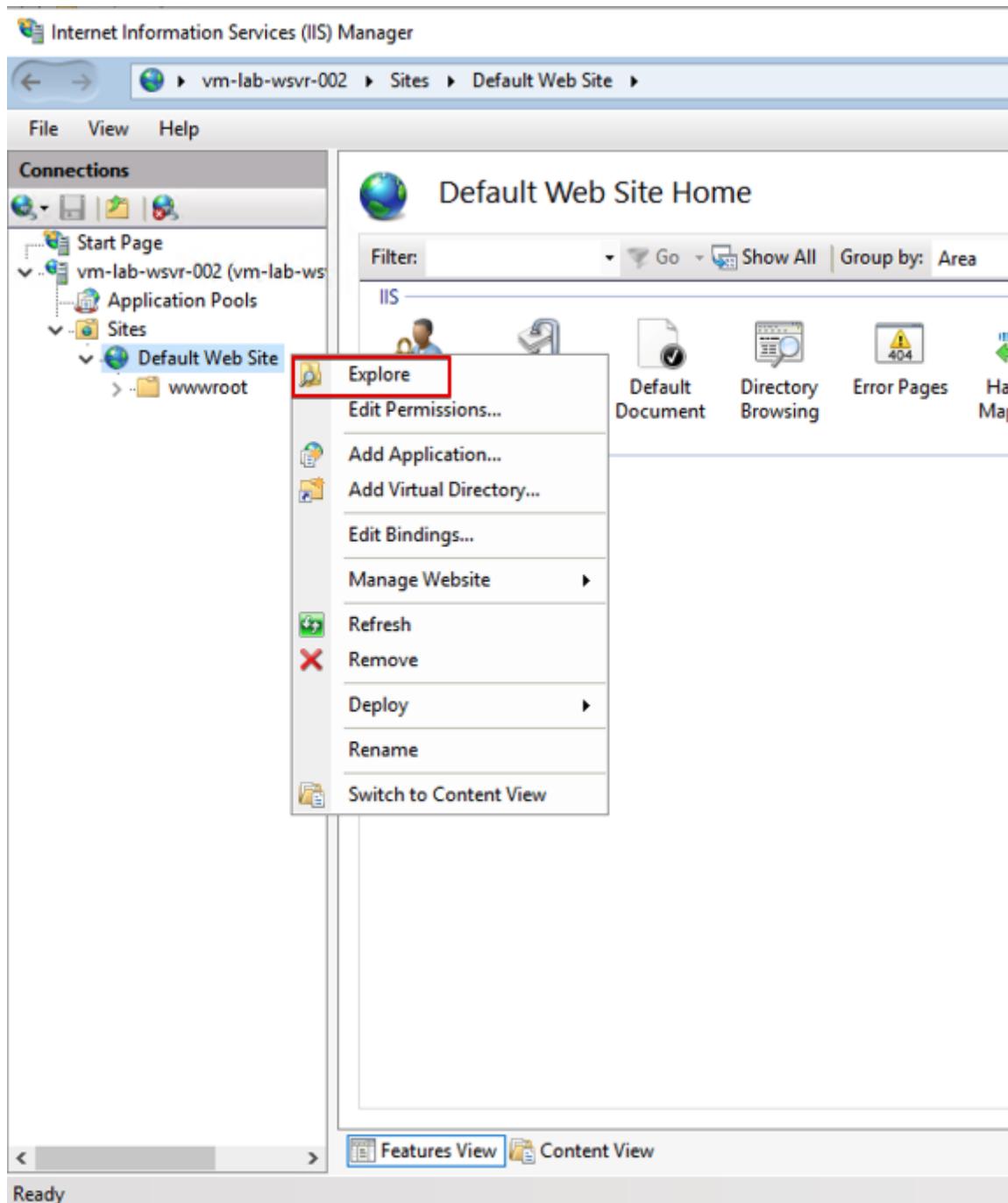
ないように**Read-only**にチェックを付けます。



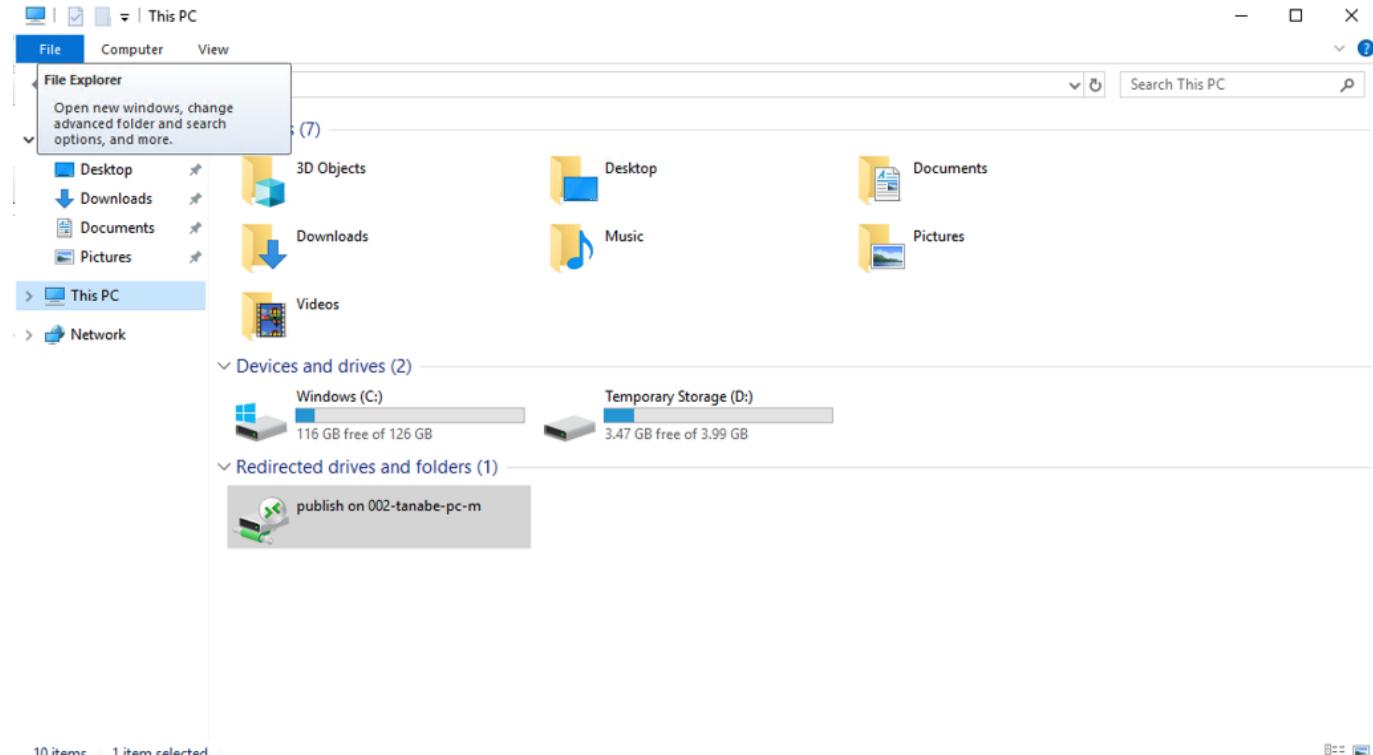
再びRDPに戻り、 Server ManagerからIIS Managerを選択します。



Default Web Siteで右クリック->**Explore**を選択します。



Macの**Microsoft Remote Desktop**で設定したフォルダがアタッチされているのが確認できました。
デプロイしたいファイル一式となります。

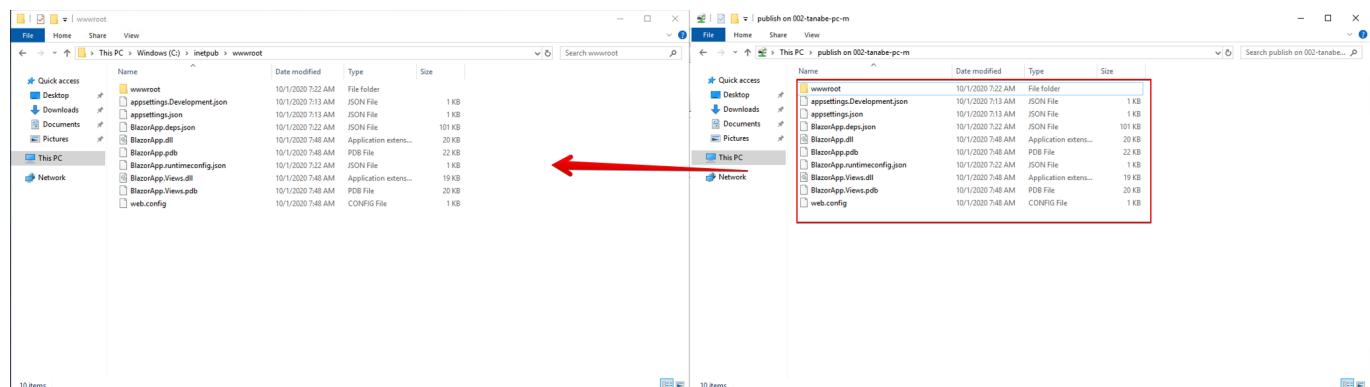


10 items | 1 item selected



Windows ServerのC:\inetpub\wwwrootまでエクスプローラーを開いて、上記手順で開いたMac内のファイルを全てコピーします。

これでデプロイが完了です。早速Webページを開いてみましょう。



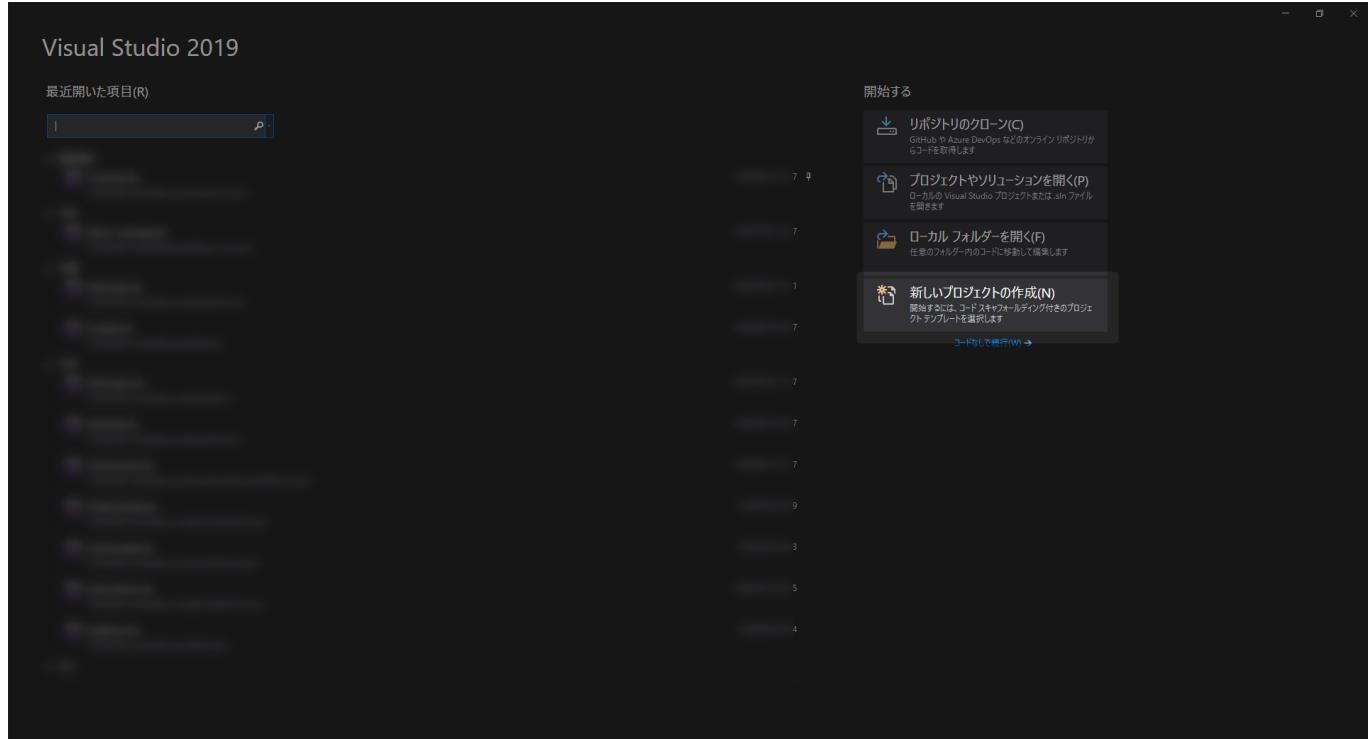
Blazor Serverで作成されたページが表示されましたね。

Macの場合は少し面倒ですが、デプロイ時は上記の手順を再度繰り返せばOKです。

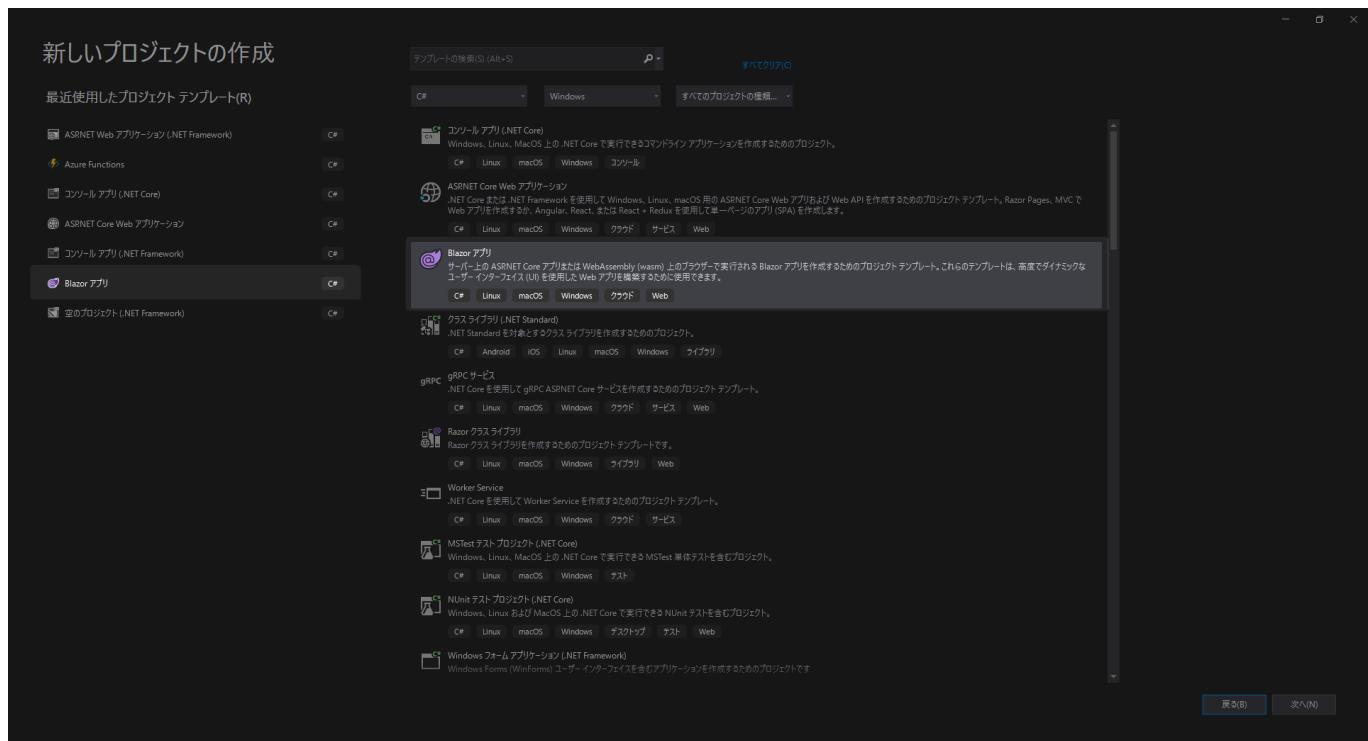
The screenshot shows a Blazor web application running in a browser. The URL is http://localhost:5000. The page has a dark blue header with the title 'BlazorApp'. On the left, there is a navigation menu with 'Home', 'Counter', and 'Fetch data' options. The main content area displays the text 'Hello, world!' in large white font, followed by the message 'Welcome to your new app.' and a survey link: 'How is Blazor working for you? Please take our [brief survey](#) and tell us what you think.'

Windowsを使用している方

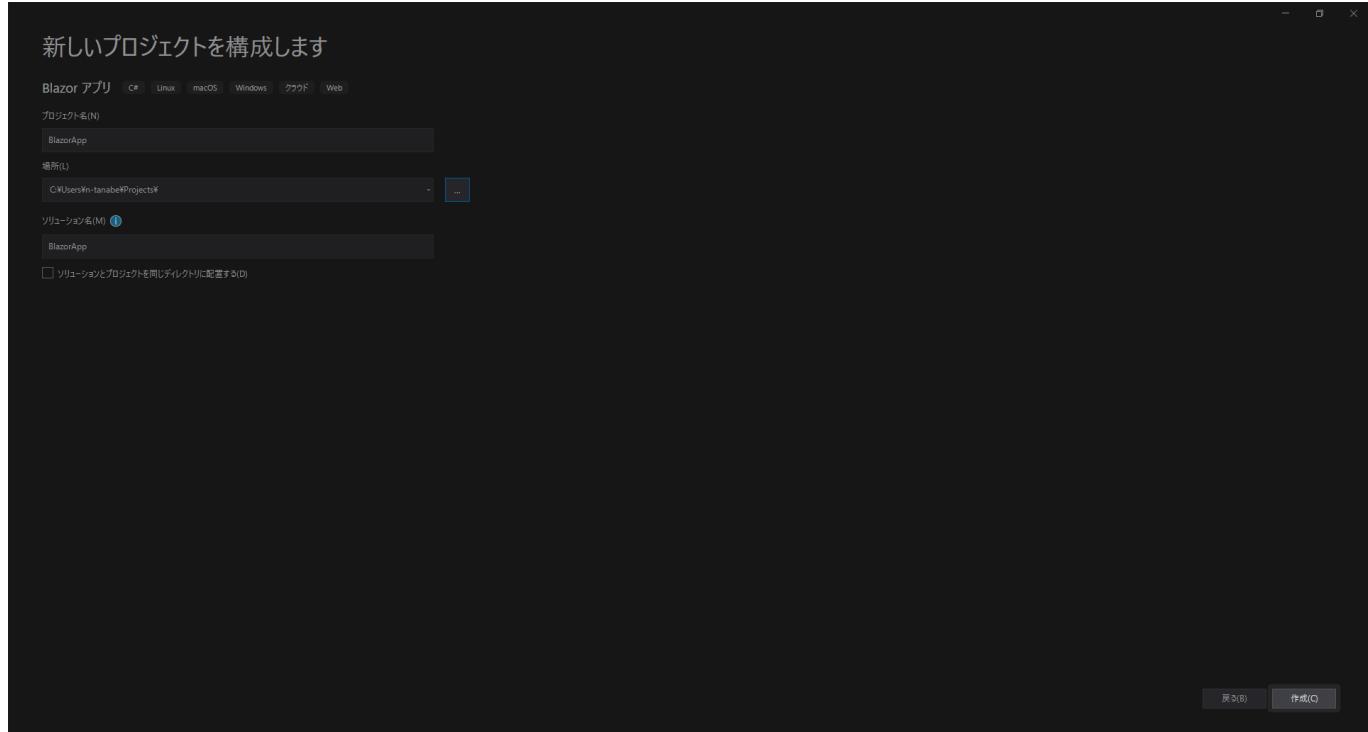
Visual Studioを開いて新規プロジェクトを作成します。



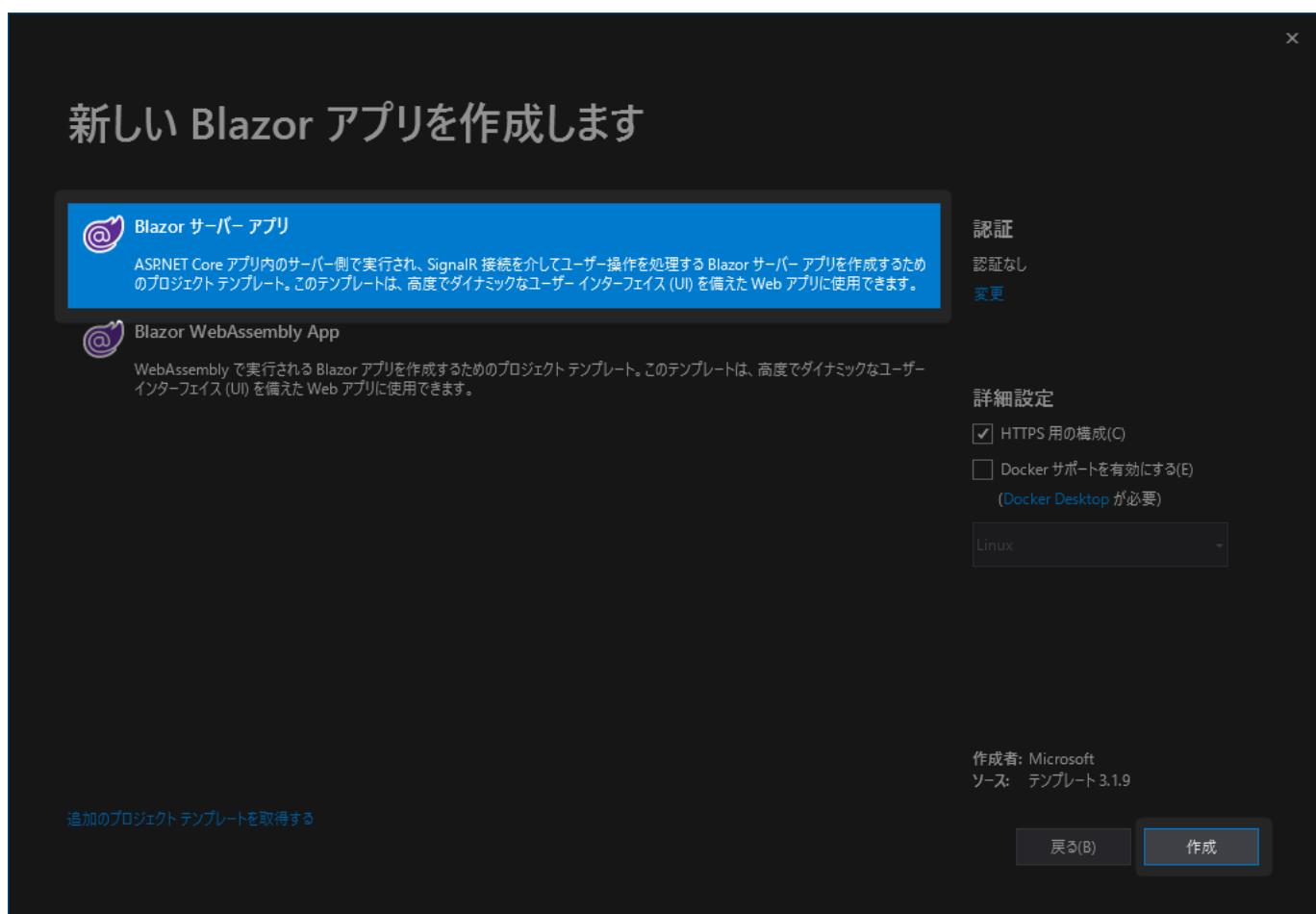
Blazorアプリを選択してください。



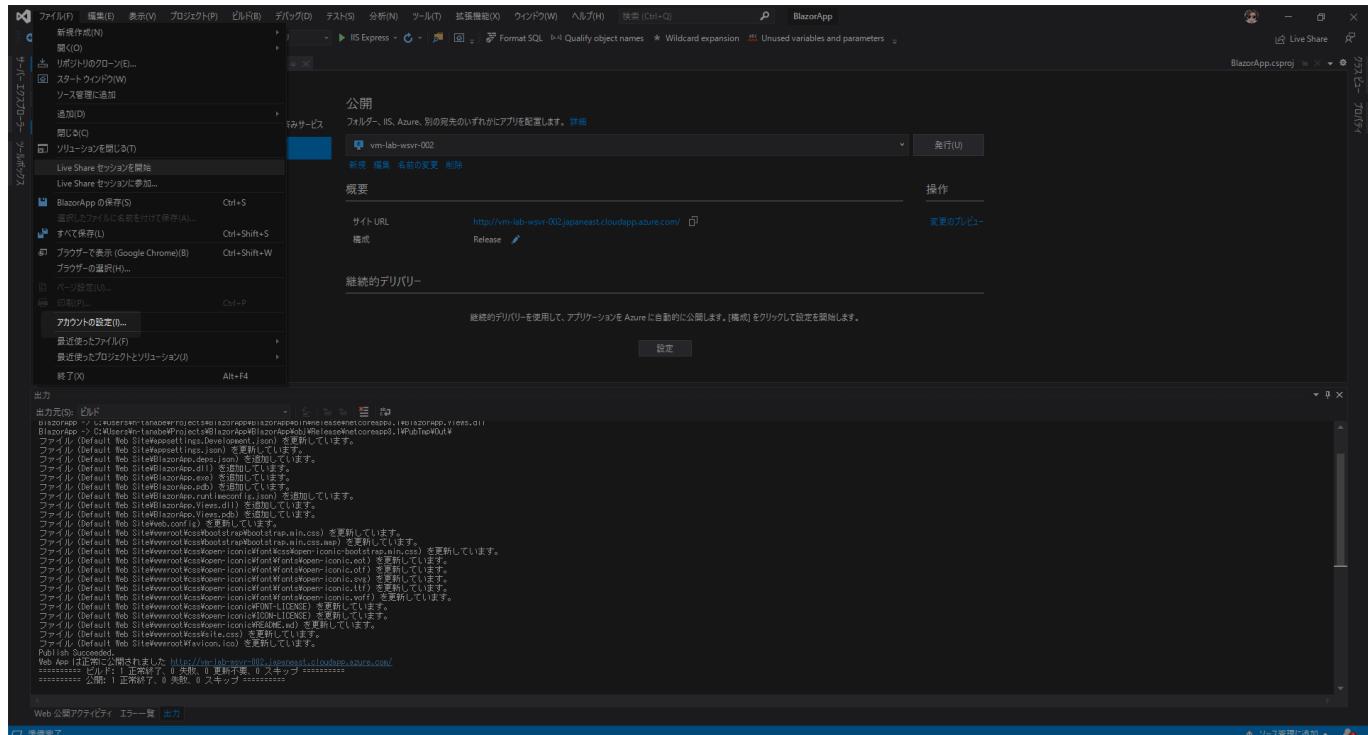
プロジェクト名や場所は任意です。



Blazorサーバーアプリを選択し、他は特に変更せず作成します。



Visual Studio上のアカウント設定から自身のAzureアカウントでログインしましょう。



個人アカウント


Tanabebe
nob@tanabebe.tech

[Visual Studio プロファイルの管理](#)

[アカウントのオプション](#)

[サインアウト](#)

個人アカウントは、複数のデバイス間での設定のローミングや、他の個人用サービスの提供に使用されます。

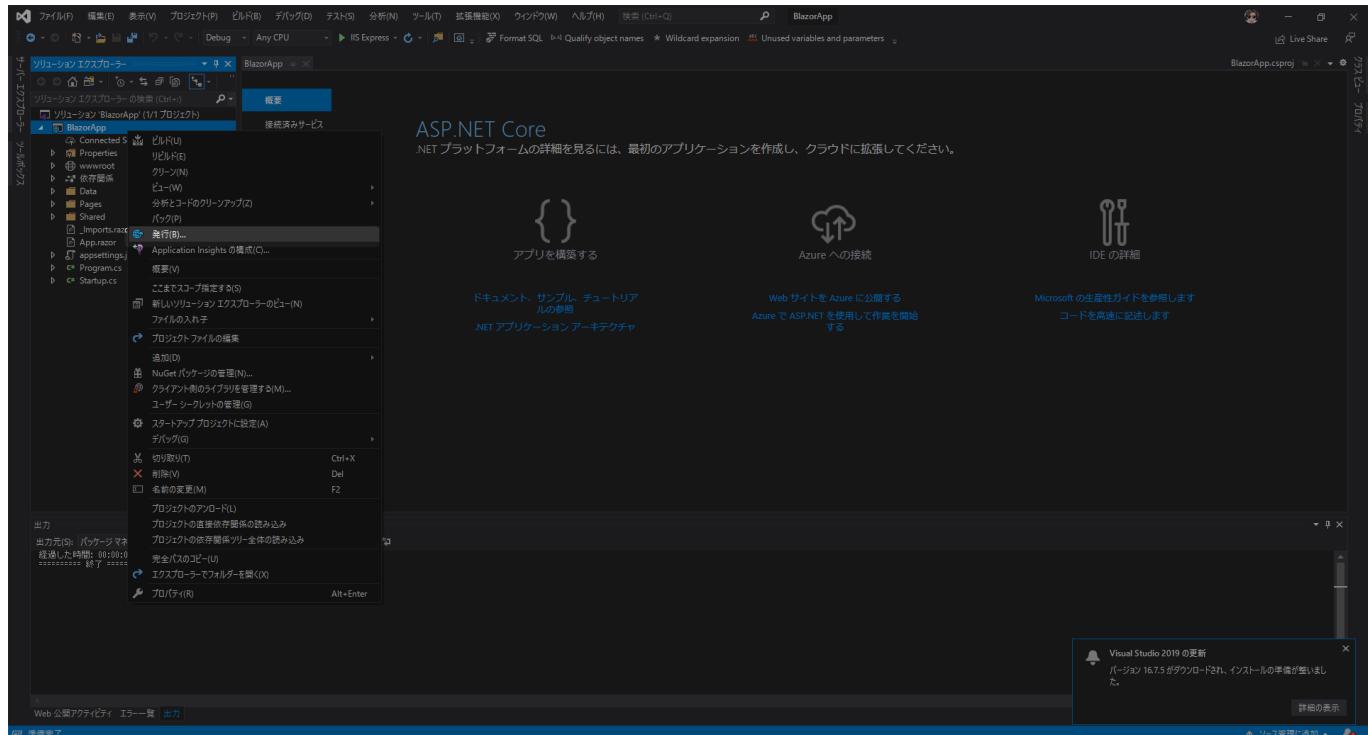
すべてのアカウント

 Microsoft account nob@tanabebe.tech	削除 フィルターの実行...
--	---

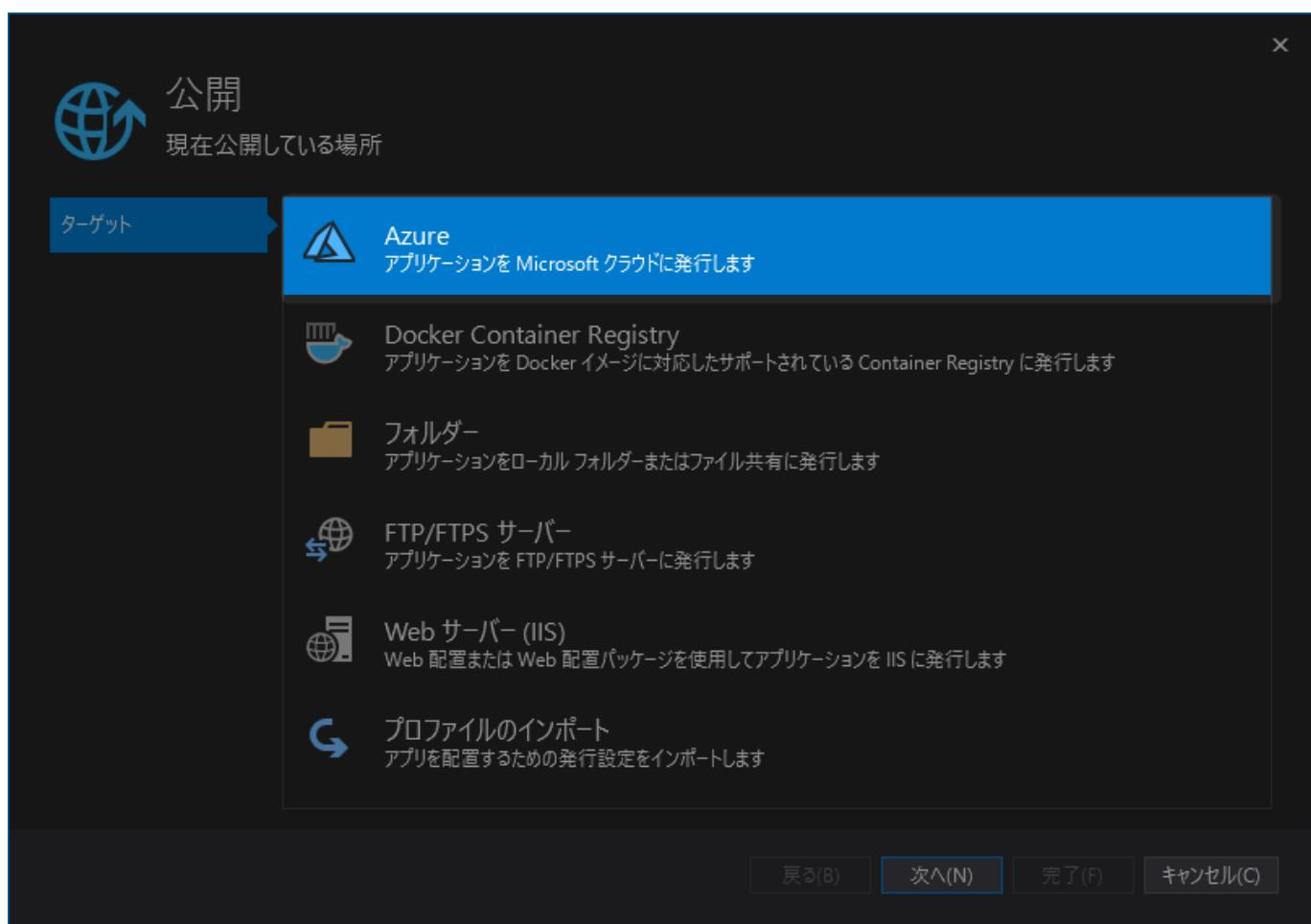
[アカウントの追加...](#)

16.0.30503.244 D16.7 閉じる(C)

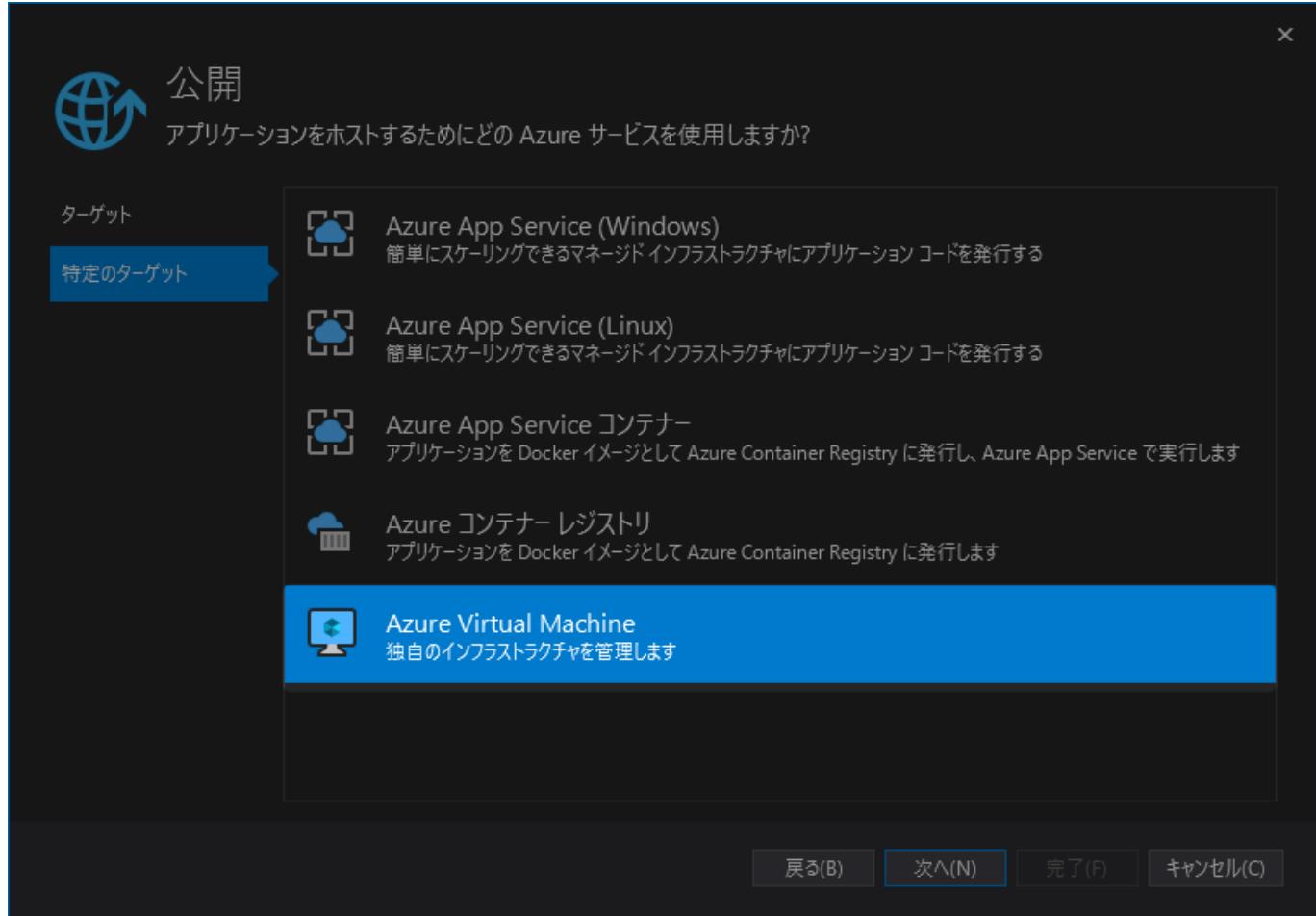
プロジェクトが作成されたらプロジェクトを右クリックし発行を選択します。



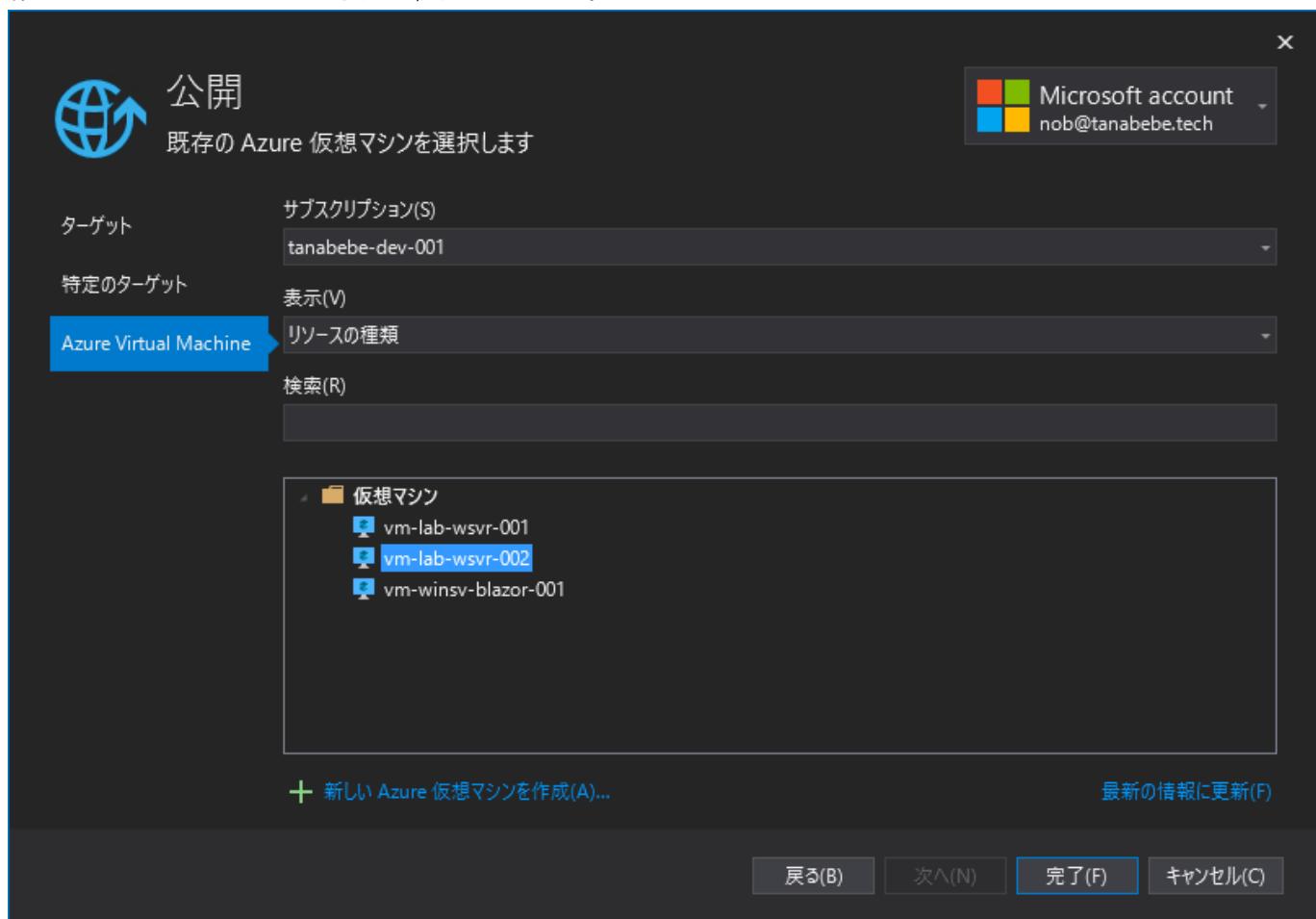
Azureを選択して次へ進みます。



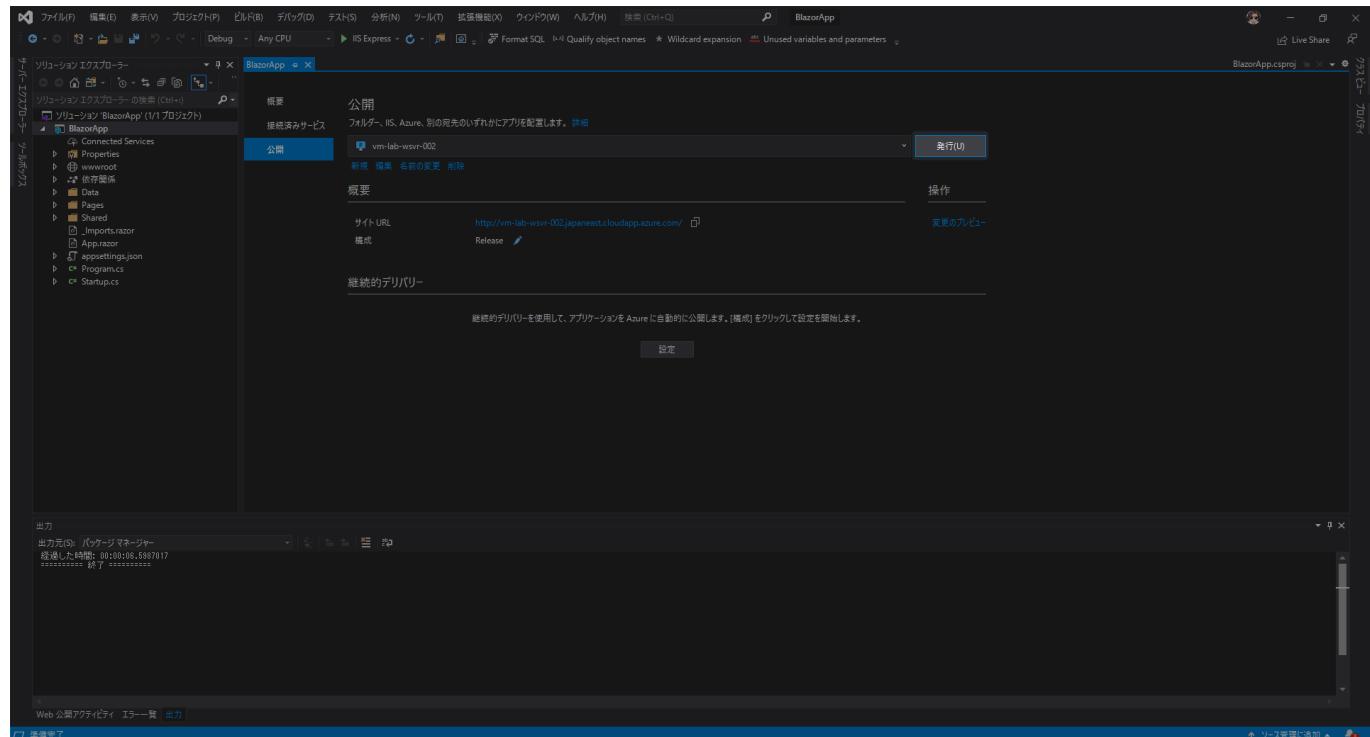
Azure Virtual Machineを選択し、次へ進みます。



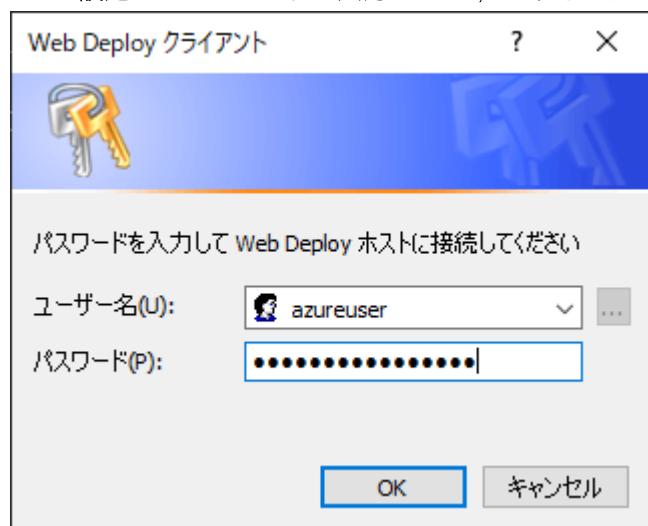
作成したVirtual Machineを選択して、完了とします。



公開用のプロファイルが作成できたので、発行をクリックします。



VMで設定したパスワードを入力してOK、とするとデプロイが始まります。



正常にデプロイが完了すると以下のとおり、Visual Studio内の出力が確認できます。

```
出力  
出力元(S): ビルド  
1>----- ビルド開始: プロジェクト: BlazorApp, 構成: Release Any CPU -----  
1>BlazorApp -> C:\Users\h-tanabe\Projects\BlazorApp\BlazorApp\bin\Release\netcoreapp3.1\BlazorApp.dll  
1>BlazorApp -> C:\Users\h-tanabe\Projects\BlazorApp\BlazorApp\bin\Release\netcoreapp3.1\BlazorApp.Views.dll  
2>----- 公開の開始: プロジェクト: BlazorApp, 構成: Release Any CPU -----  
BlazorApp -> C:\Users\h-tanabe\Projects\BlazorApp\BlazorApp\bin\Release\netcoreapp3.1\BlazorApp.dll  
BlazorApp -> C:\Users\h-tanabe\Projects\BlazorApp\BlazorApp\bin\Release\netcoreapp3.1\BlazorApp.Views.dll  
BlazorApp -> C:\Users\h-tanabe\Projects\BlazorApp\BlazorApp\obj\Release\netcoreapp3.1\PubTmp\Out\  
ファイル (Default Web Site\appsettings.Development.json) を更新しています。  
ファイル (Default Web Site\appsettings.json) を更新しています。  
ファイル (Default Web Site\BlazorApp.deps.json) を追加しています。  
ファイル (Default Web Site\BlazorApp.dll) を追加しています。  
ファイル (Default Web Site\BlazorApp.exe) を追加しています。  
ファイル (Default Web Site\BlazorApp.pdb) を追加しています。  
ファイル (Default Web Site\BlazorApp.runtimeconfig.json) を追加しています。  
ファイル (Default Web Site\BlazorApp.Views.dll) を追加しています。  
ファイル (Default Web Site\BlazorApp.Views.pdb) を追加しています。  
ファイル (Default Web Site\web.config) を更新しています。  
ファイル (Default Web Site\wwwroot\css\bootstrap\bootstrap.min.css) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\font\css\open-iconic-bootstrap.min.css.map) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\font\css\open-iconic.eot) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\font\fonts\open-iconic.otf) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\font\font\fonts\open-iconic.svg) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\font\fonts\open-iconic.ttf) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\font\fonts\open-iconic.woff) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\font\LICENSE) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\ICON-LICENSE) を更新しています。  
ファイル (Default Web Site\wwwroot\css\open-iconic\README.md) を更新しています。  
ファイル (Default Web Site\wwwroot\css\site.css) を更新しています。  
ファイル (Default Web Site\wwwroot\favicon.ico) を更新しています。  
Publish Succeeded.  
Web App は正常に公開されました http://vm-lab-wsvr-002.japaneast.cloudapp.azure.com/  
===== ビルド: 1 正常終了、0 失敗、0 更新不要、0 スキップ ======  
===== 公開: 1 正常終了、0 失敗、0 スキップ ======
```

正常に完了したら早速Webページを開いてみましょう。

