Publishing Application to Azure VM

In the previous lab, we connect our web-based application to the cloud using Azure Storage Account ran our application locally using our laptop, this time we will run our application on our Virtual Machine using Internet Information Services. In previous labs we have already talked about Azure Virtual Machines and in this lab, we will talk about IIS.

Internet Information Services (IIS) is a web server developed by Microsoft, designed for the hosting of web applications and services. In the context of Azure, IIS does not function as an independent service; rather, it is installed on a virtual machine (VM) that operates on a Windows Server operating system.

Key Features of IIS's

1. **Web Server Role**: IIS provides a platform to host and manage websites, web applications, and services. Supports technologies like **ASP.NET Core**, **PHP**, **HTML**, and **JavaScript**.
2. **Customizable Infrastructure**: IIS runs on an Azure VM, giving you full control over the environment (e.g., installing software, configuring settings). Useful for scenarios requiring custom configurations not available in Azure PaaS offerings.
3. **Supports Various Deployment Models**: You can deploy applications to IIS using tools like **Visual Studio**, **Web Deploy**, or manual configurations.
4. **Integration with Azure Services**: IIS on Azure can be integrated with **Azure SQL Database**, **Blob Storage**, and **Azure Monitor** for end-to-end application management.
5. **Scalability**: You can scale the Azure VM hosting IIS vertically (increasing resources) or horizontally (adding more VMs).

Use Cases of IIS in Azure

1. **Hosting Custom Web Applications:** ASP.NET, PHP, or Node.js applications requiring custom IIS settings.
2. **Migrating On-Premises IIS Applications:** Easily migrate applications from on-premises IIS servers to Azure VMs.
3. **Testing and Development:** Use IIS to replicate on-premises environments for development and testing purposes.

The end goal is to host your .NET application on an Azure Virtual Machine, making it accessible via a public DNS name. By configuring the Azure VM with IIS, the required management service, .NET 6.0 runtime, and Web Deploy, you create a hosting environment compatible with your application. This includes setting up network security rules (e.g., port 8172) and a unique DNS name. Finally, publishing the application from Visual Studio deploys it to the IIS server on the VM, allowing users to access the application through a web browser by using the VM's DNS name. This demonstrates hosting on Azure's IaaS platform.

To begin with the lab

1. Firstly, you need to download the RDP file of your VM that you have created and log into it.

2. Initially, we will install IIS within our virtual machine, followed by the installation of the management service, which will enable us to execute our application. Additionally, we will verify that .NET 6 is operational on our server, as our project is dependent on it.
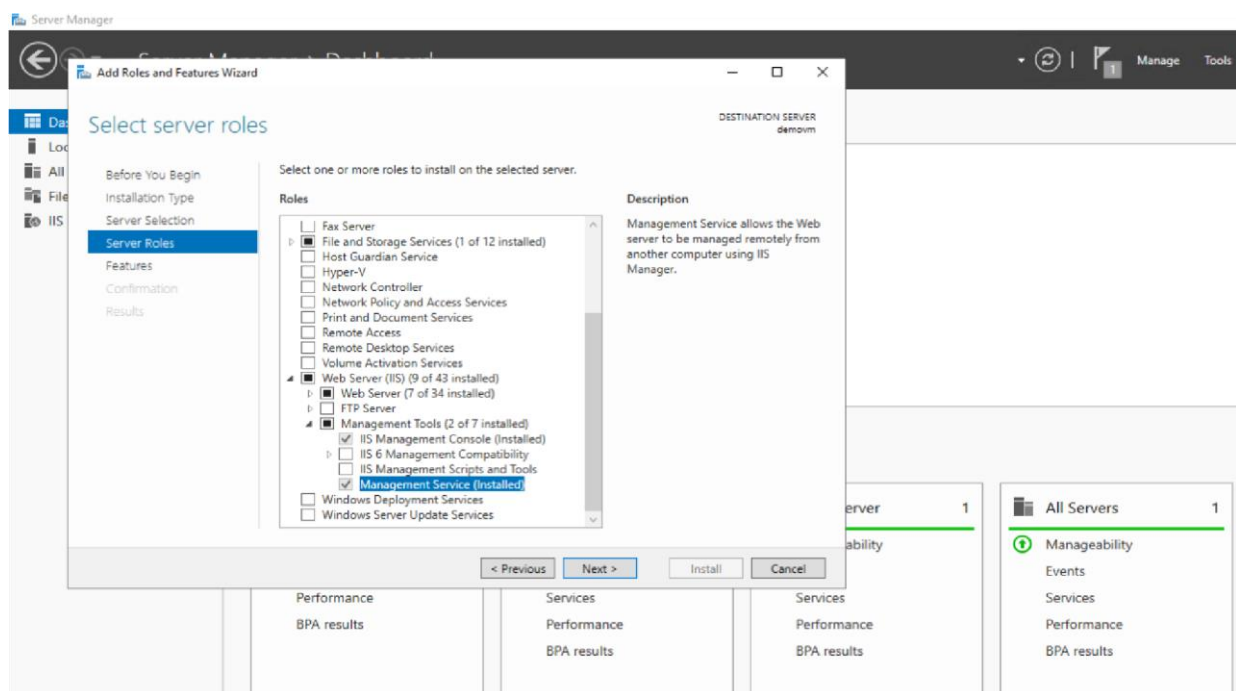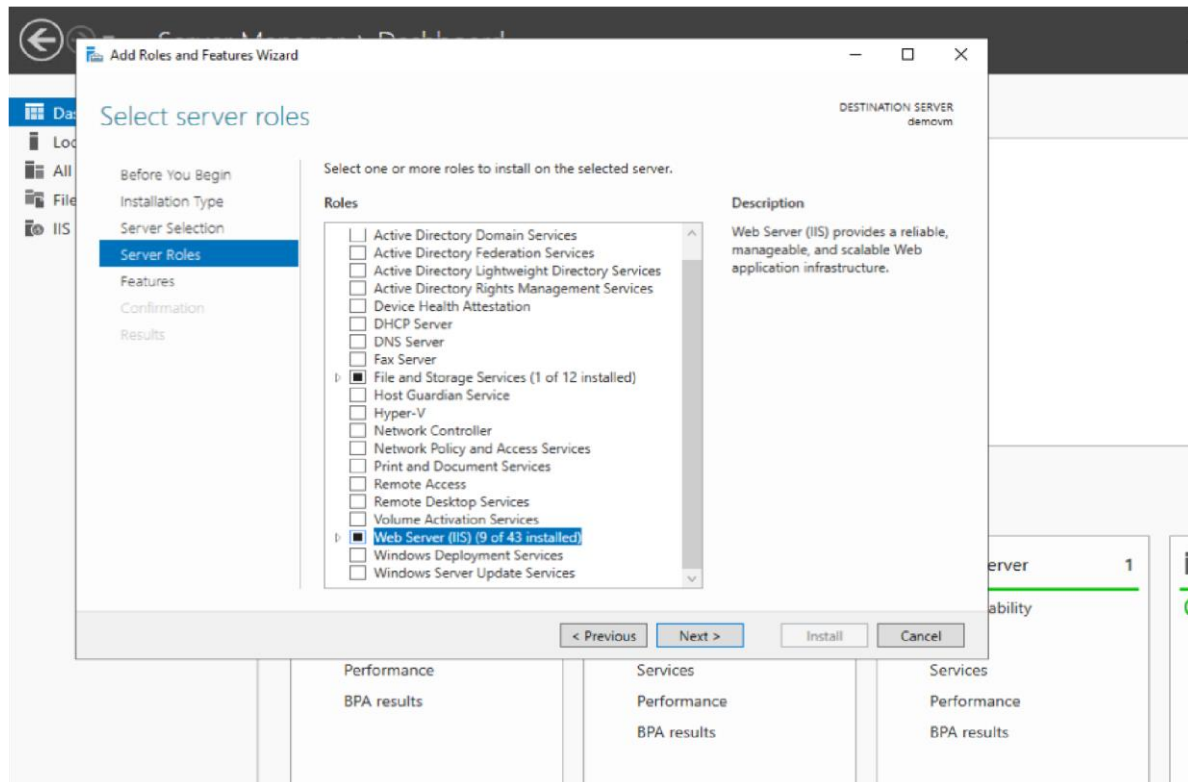
3. After the installation is complete, it is necessary to access the local servers and disable Internet Explorer enhanced security. This action will allow us to download and install applications from the internet on our virtual machine.
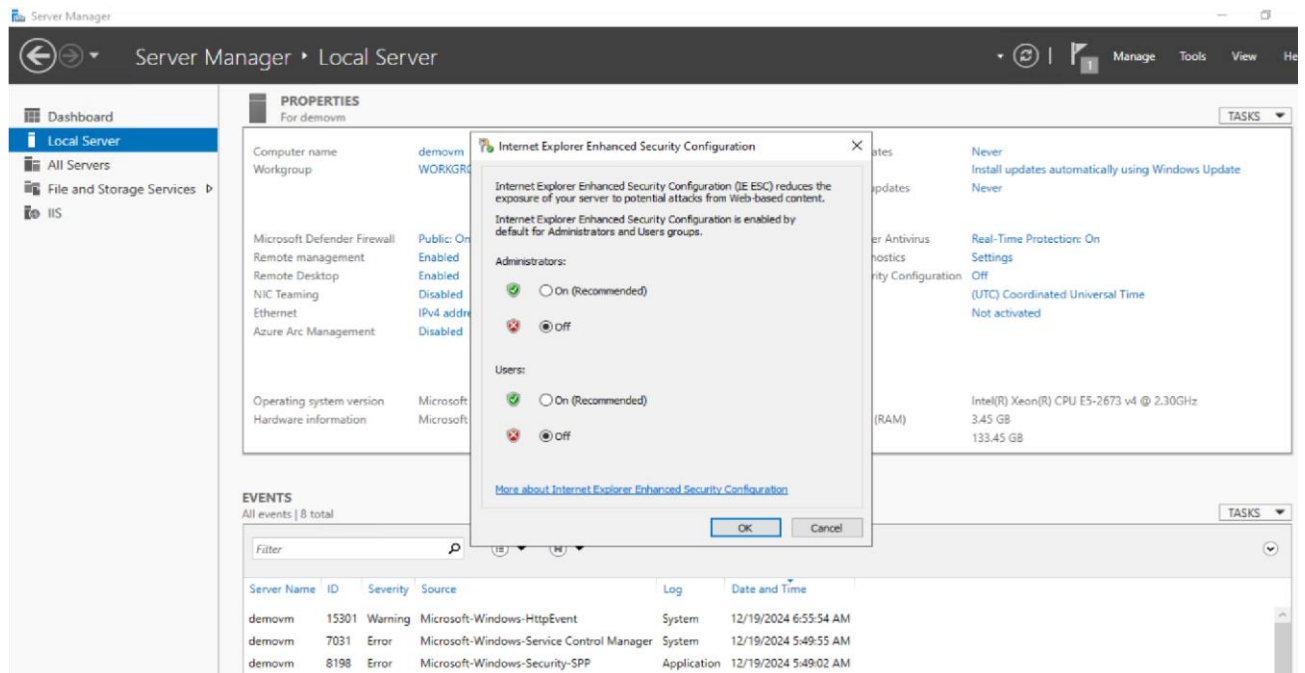
4. Now launch the Edge browser and conduct a search for .NET 6 in order to download it. Subsequently, proceed to download the Windows Hosting bundle. Once the download is complete, you may then install it.
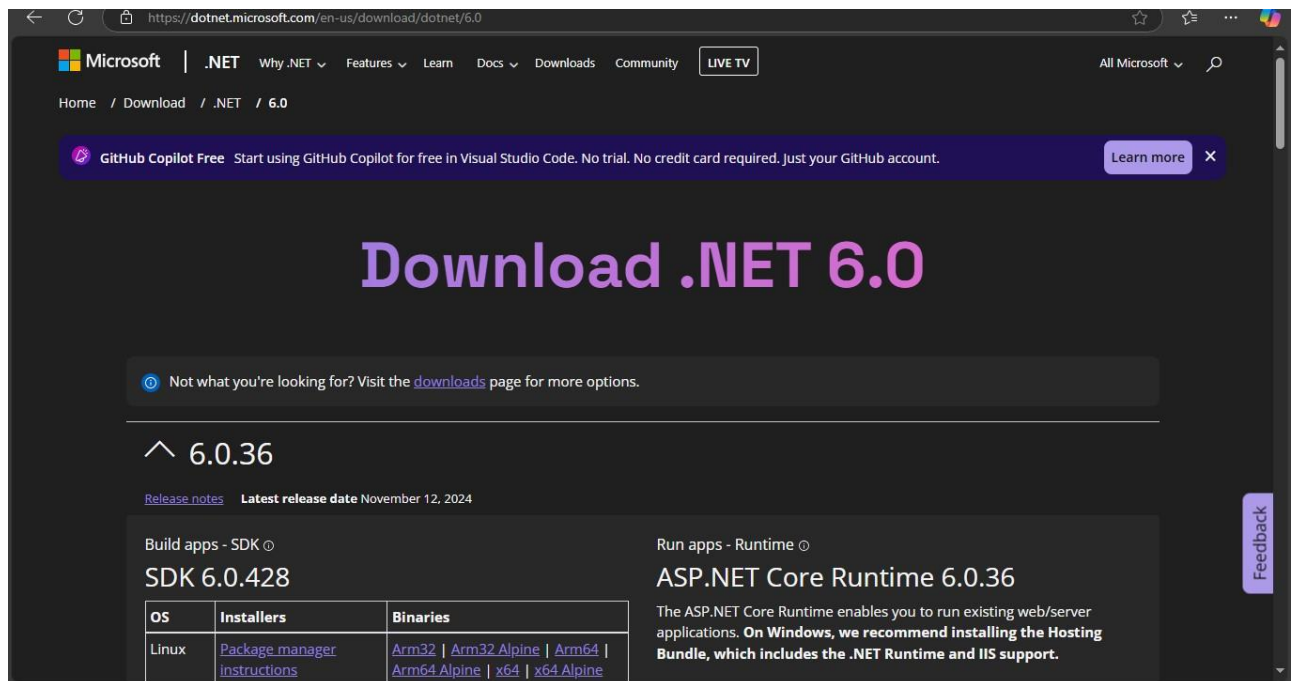
Run apps - Runtime ⓘ

ASP.NET Core Runtime 6.0.36

The ASP.NET Core Runtime enables you to run existing web/server applications. **On Windows, we recommend installing the Hosting Bundle, which includes the .NET Runtime and IIS support.**

IIS runtime support (ASP.NET Core Module v2)
16.0.24290.36

OS	Installers	Binaries
Linux	Package manager instructions	Arm32 | Arm32 Alpine | Arm64 | Arm64 Alpine | x64 | x64 Alpine
macOS		Arm64 | x64
Windows	x64 | x86 | Hosting Bundle | winget instructions	x64 | x86 | Arm64

5. To proceed, you must locate the web deploy tool and download it from the official Microsoft website. Ensure that you download the file named (webdeploy_amd64_en-US.msi). Following the download, simply proceed with the installation.

Web Deploy v4.0

The Web Deployment Tool simplifies migration, management and deployment of IIS Web servers, Web applications and Web sites.
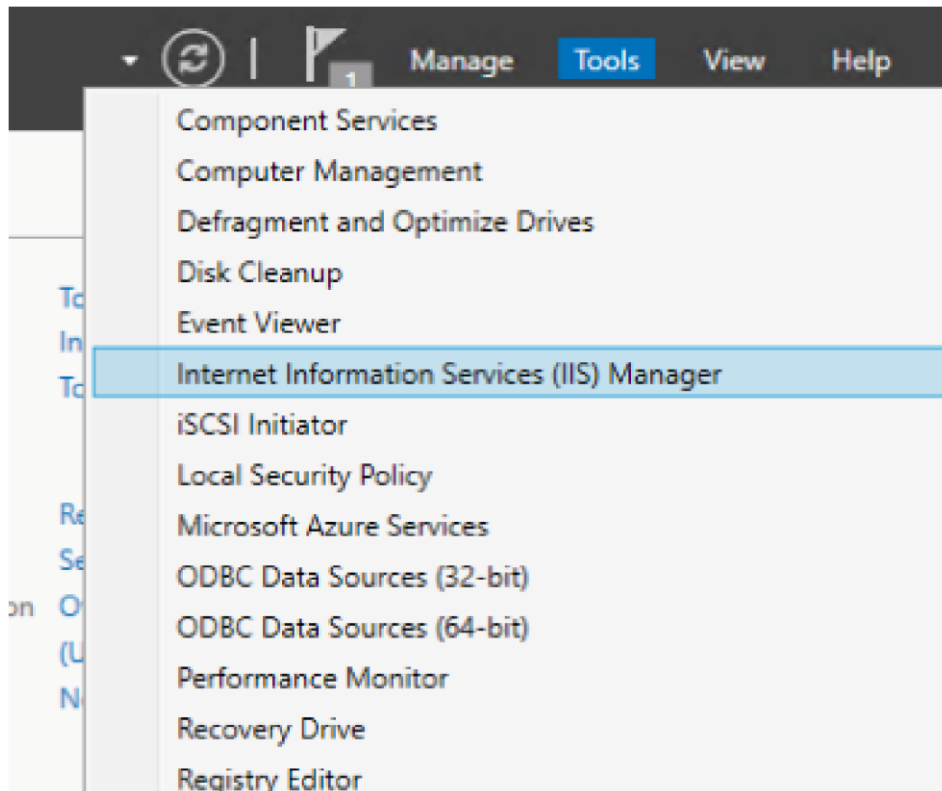
Important! Selecting a language below will dynamically change the complete page content to that language.
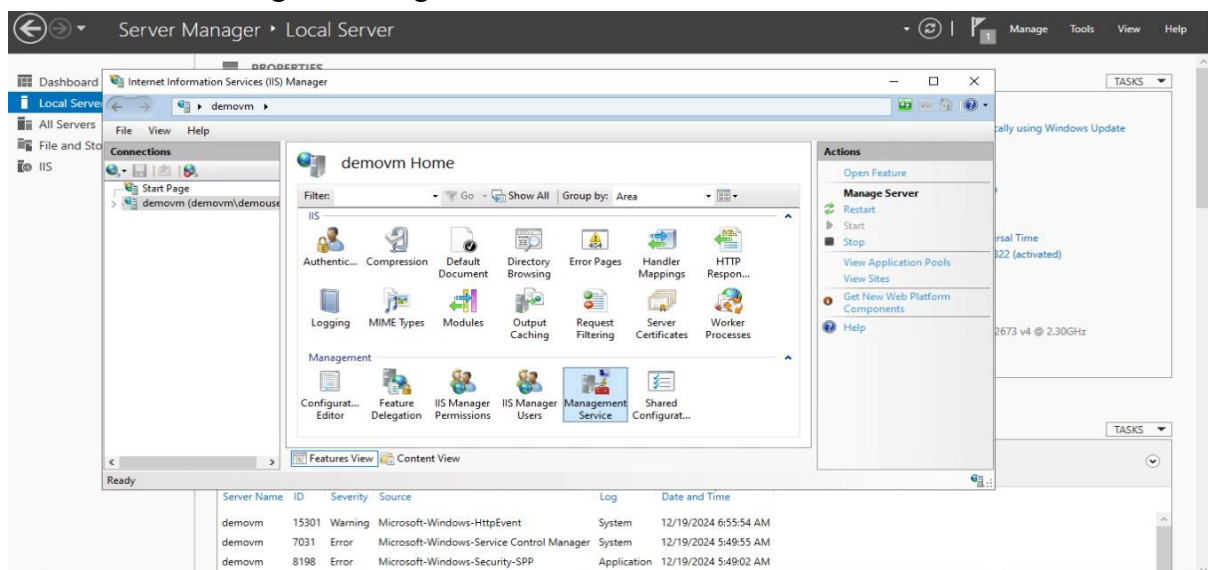
Select language | English ∨ | **Download**

6. Once both of the above services are downloaded install them and move to the next step.
7. Now you need to go to server manager and from tools go to IIS manager.

8. Now we will go to management services.

9. To ensure that the remote connection is enabled, it is necessary to first stop this service. Once this is completed, restart the service and additionally, make sure to include port 8172 in the Network Security Group (NSG) of your virtual machine. Also, make sure that enable remote connections options is having the tick mark in the check box.
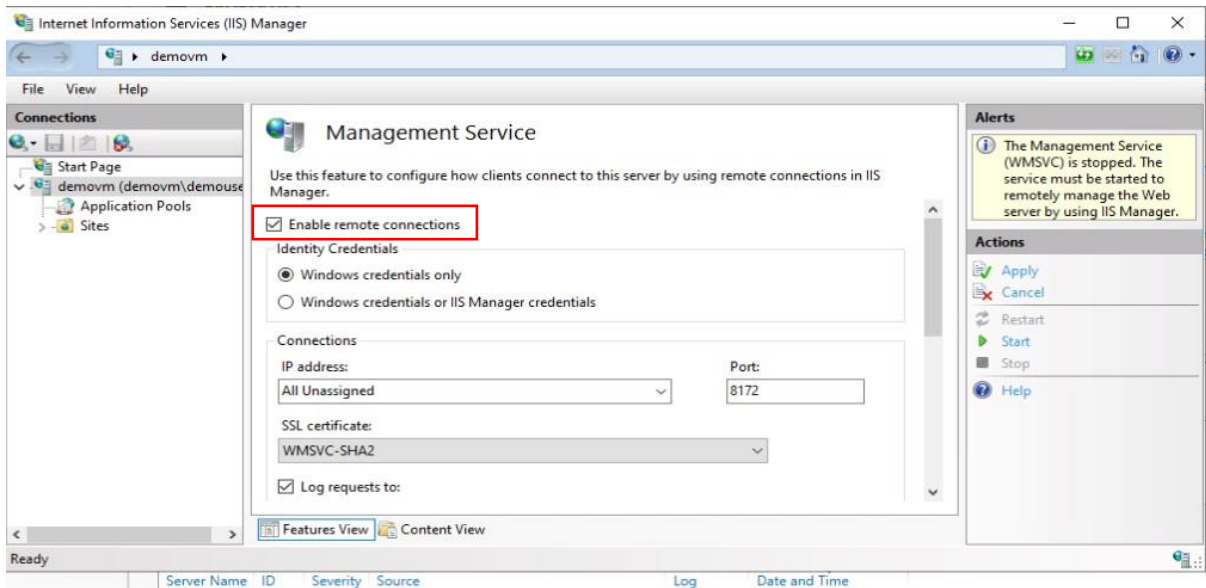
10. Now go to your Azure Virtual Machine in Azure portal and in NSG add port 8172 in your inbound rules. Also, add the HTTP port 80.
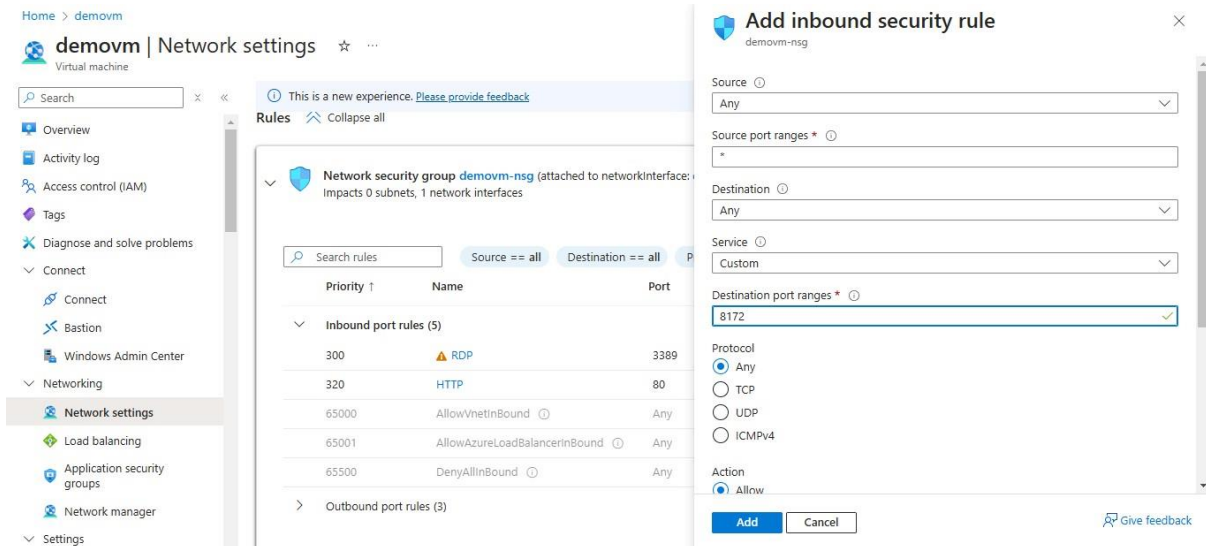
11. Now go to overview in left pane of your Virtual Machine and configure DNS name from there. Give a unique name to your DNS.
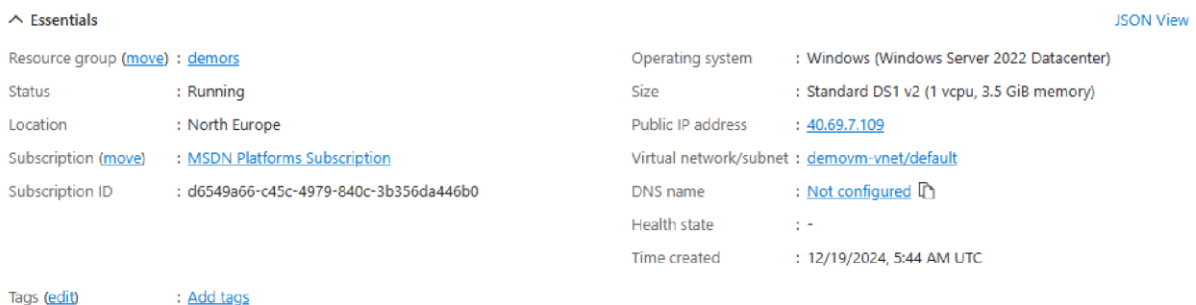
IP address assignment
Static

IP address ⓘ
40.69.7.109

Idle timeout (minutes) ⓘ

○── | 4 |

DNS name label (optional) ⓘ

| demovm   ✓ |

.northeurope.cloudapp.azure.com

ⓘ You can use the IP address as your 'A' DNS record or DNS label as your 'CNAME' record. Learn more about adding a custom domain to this IP address ↗
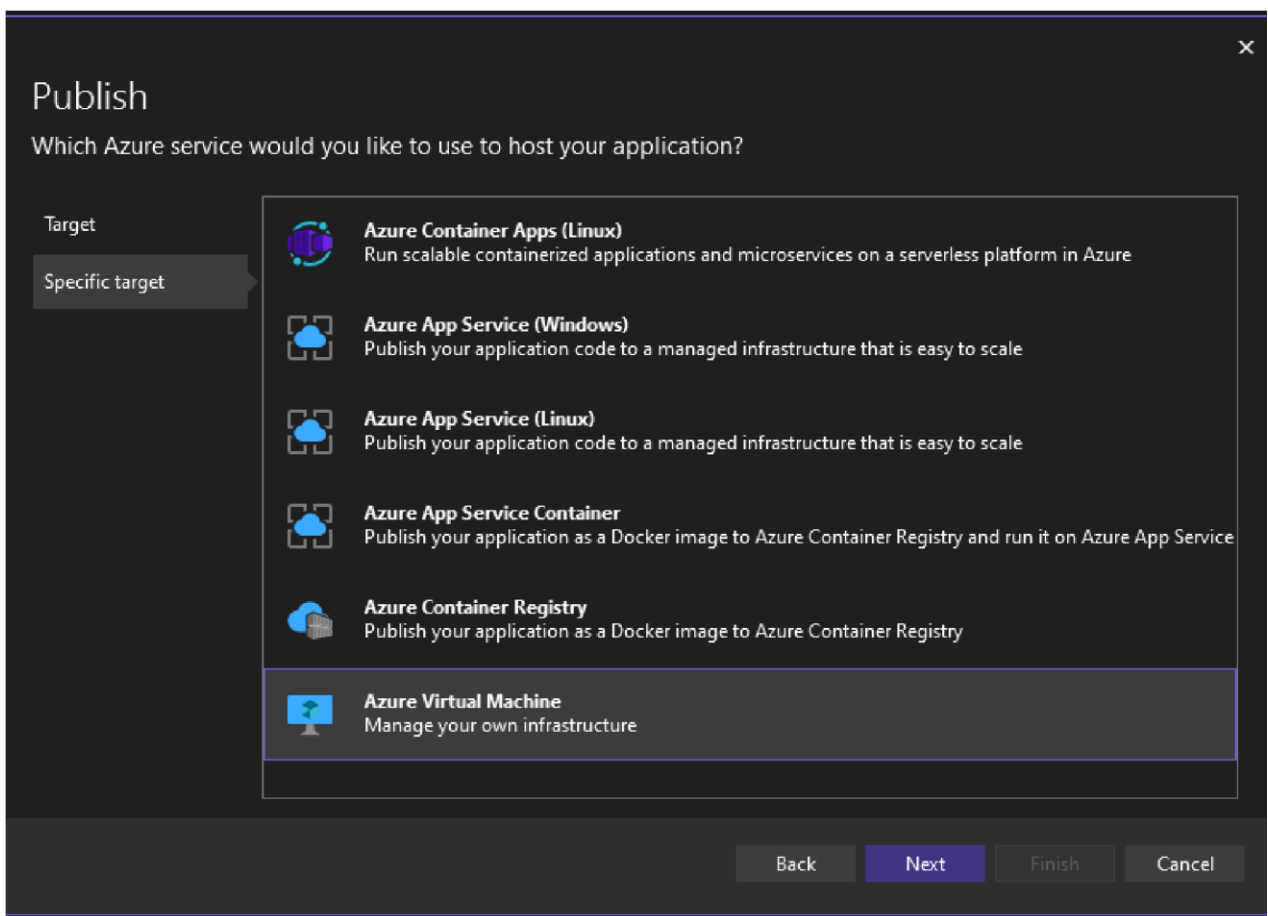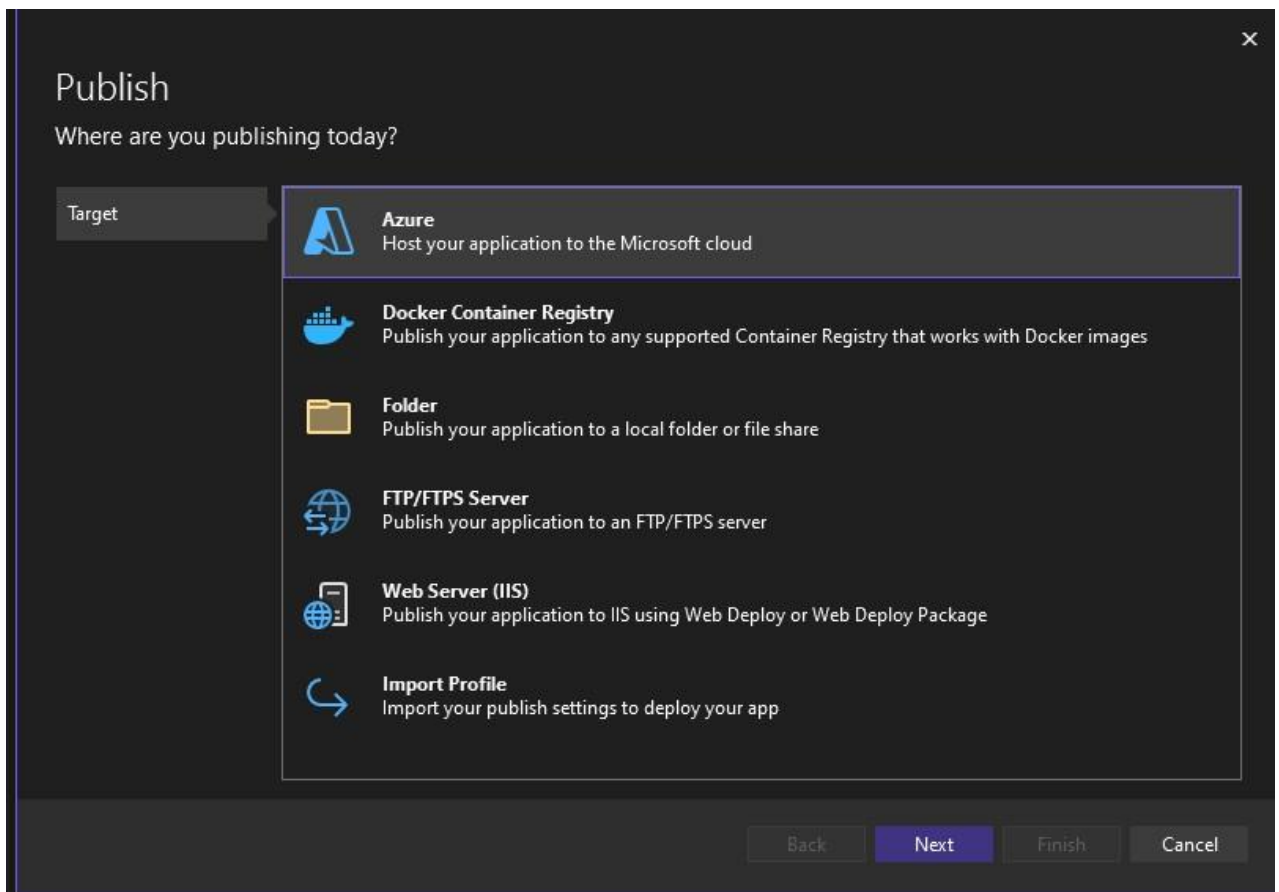
Alias record sets
Create an alias record in Azure DNS. Learn more ↗
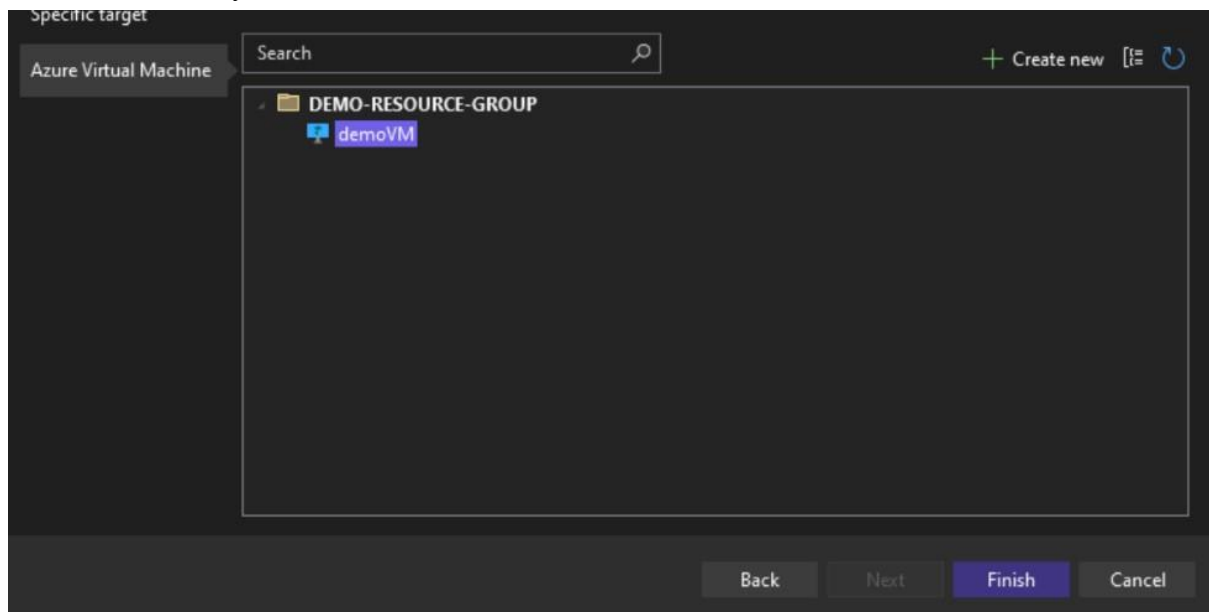+ Create alias record

12. Now navigate to your Visual Studio environment where your application is located, right-click on the application, and select the option to publish. Select Azure and then Azure VM.

Publish

Where are you publishing today?

Target

Azure
Host your application to the Microsoft cloud

Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images

Folder
Publish your application to a local folder or file share

FTP/FTPS Server
Publish your application to an FTP/FTPS server

Web Server (IIS)
Publish your application to IIS using Web Deploy or Web Deploy Package

Import Profile
Import your publish settings to deploy your app

Back   Next   Finish   Cancel

Publish

Which Azure service would you like to use to host your application?

Target

Specific target

Azure Container Apps (Linux)
Run scalable containerized applications and microservices on a serverless platform in Azure

Azure App Service (Windows)
Publish your application code to a managed infrastructure that is easy to scale

Azure App Service (Linux)
Publish your application code to a managed infrastructure that is easy to scale

Azure App Service Container
Publish your application as a Docker image to Azure Container Registry and run it on Azure App Service

Azure Container Registry
Publish your application as a Docker image to Azure Container Registry

Azure Virtual Machine
Manage your own infrastructure

Back   Next   Finish   Cancel

13. Select your Azure Virtual Machine.

14. Our project has not yet been published. To proceed, we must select "More Actions" and then choose "Edit." Next, navigate to the "Connection" section, input the password for your virtual machine, and validate the connection before saving it.

15. Once this step is completed, we can click the "Publish" button to make our project live. Subsequently, copy the DNS name of your virtual machine and paste it into a new browser window to view your web page in operation.