What is azure app service

App Service is a Platform as a Service (PaaS) offering from Microsoft. We use it to host web applications, REST API's and backend services for mobile applications.

It doesn't really matter which programming language or framework you have used. Web applications and services that are developed using any of the following programming languages or frameworks can be hosted using azure app service. It could be

* .NET
* .NET Core
* Java
* Ruby
* Node.js
* PHP
* Python

Azure App Service v/s On-premise Hosting

How is hosting on azure app service different from hosting on your own on-premise server. In other words, why would anyone use azure app service instead of using other alternatives to host their applications.

With on-premise web hosting, you or your organisation is reponsible for managing pretty much everything.

* You may have to spec out and procure physical servers, storage, networking equipment and all the related hardware.
* Make sure there is main power supply, back-up power supply, cooling system etc are in place.
* Install and set up the network
* Install and configure virtualization software, operating system, any middleware or runtime components that your application needs
* Install and configure a web server like IIS, Apache, Nginx etc.

Azure App Service is a Platform as a Service (PaaS) offering. This means you or your organization is only responsible for managing your business application and it's data. Everything else is managed by Azure. You don't have to worry about any of the things like, managing the network or underlying infrastructure. Installing the operating system updates, critical patches, runtime or middleware components. All these are taken care by Azure. This gives you, even more time to concentrate on what matters to your business.

Benefits of using azure app service

Fully managed environment

It's a fully managed environment, meaning App Service automatically patches and maintains the OS and language frameworks for you. You get the time to focus on designing, developing and maintaining your application and data.

Multiple programming languages and frameworks are supported

Azure App Service supports a wide variety of programming languages and frameworks.

* .NET
* .NET Core
* Java
* Ruby
* Node.js
* PHP
* Python

You can also run PowerShell and other scripts or executables as background services.

Scalability

Based on the demand for your application, App Service can scale resources up and down or in and out. You can do this either manually if you want to or automatically based on metrics like CPU utilization for example.

Compliance

App Service is ISO (International Organization for Standardization), SOC (Service Organization Controls), and PCI (Payment Card Industry) compliant.

Security

Authenticate users with Azure Active Directory or any of the external authentication providers like Google, Facebook, Twitter, or Microsoft.

Support for Containerization and Docker

You can also host a custom Windows or Linux container in App Service. So, if you want to, you can dockerize your app and host it in App Service. You can also run multi-container apps with Docker Compose. We will discuss how to do all these in our upcoming videos.

DevOps optimization

Set up CI/CD i.e continuous integration and deployment with Azure DevOps, GitHub, BitBucket, Docker Hub, or Azure Container Registry.

Access on-premises data

With App Service you can still access data on your on-premise servers using Hybrid Connections and Azure Virtual Networks.

Azure app service plan

App Service Plan

What is App Service

App Service is a Platform as a Service (PaaS) offering and we use it to host web applications, REST API's and backend services for mobile applications.

What is App Service Plan

To create App Service, you need an App Service Plan. Without an App Service Plan you cannot create App Service. So, in the Azure portal, when you try to create app service, you will have to select an app service plan if you have one already or create a new one. The point is, without an App Service Plan, you cannot create App Service. This is beacuse it is the App Service Plan that defines the compute resources required for your application to run.

How to create App Service Plan

Well, just like any other resource in azure. Search for App Service Plans in the azure protal and click Create.

On Create App Service Plan screen, you specify the following

Name

A name for the App Service Plan. Common convetion is to use the prefix plan. I am going to use this plan for our web application FirstDemo, so it makes sense to name it plan-firstdemo.

Operating System

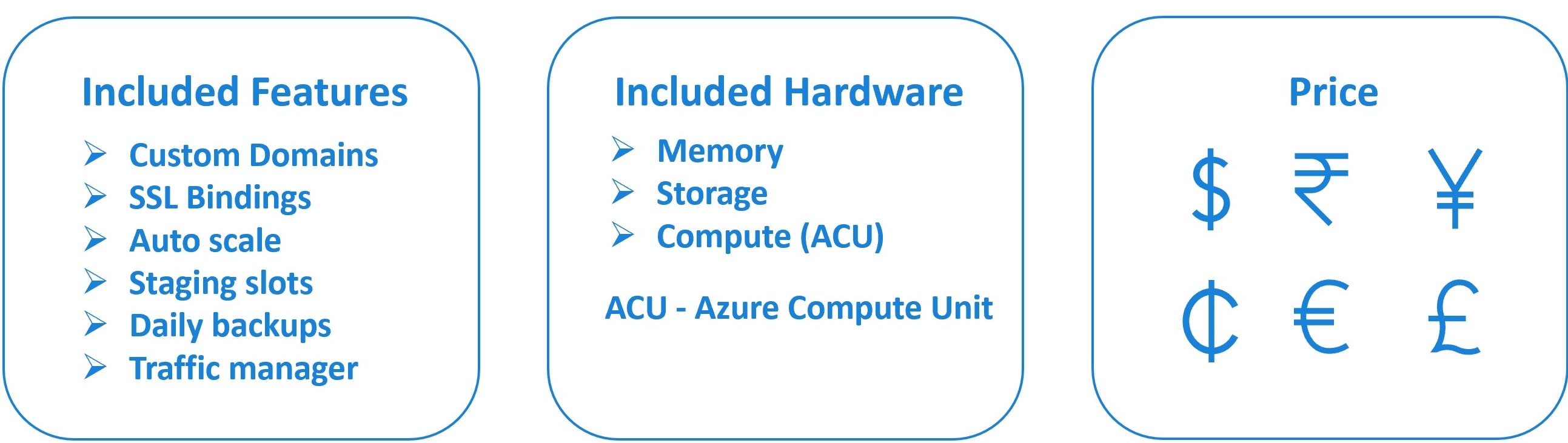
The operating system that you want on the underlying virtual machine - Windows or Linux.

Region

The region where you want the App Service Plan to be created.

Pricing Tier

It is the pricing tier that determines what you pay and what you get. Basically the pricing tier you select, determines the following 3 things.



Features available

For example the following are some of the features.

* Custom Domains
* SSL Bindings
* Auto scale
* Staging slots
* Daily backups
* Traffic manager integration

Which of these features are available to your application depends on the pricing tier you select. If you are new to these features and why we need them, please don't worry, we will discuss them in our upcoming videos. For now, what you need to understand is, the features that are available depends on the pricing tier you select. In general, the higher the tier, the more features are available and obviously you also pay more.

For example, if you select S1 pricing tier under production workload, you have all of the above features available. On the other hand, if you select the Free tier from Dev/Test workload, you don't have any of these features available. For example, if you want to be able to scale resources up and down depending on the demand for your application, you may want to select a pricing tier that supports auto scale feature. With the free pricing tier, you cannot auto scale.

Included hardware

The hardware resources that are available (like memory, storage and the compute power) also depend on the pricing tier you select. One thing I should point out here is, in Azure compute (i.e CPU) performance is compared using something called Azure Compute Unit (ACU for short).

For example, as of this course recording, if you select S1 pricing tier under production workload, you have the following hardware resources available.

* 100 Azure Compute Units
* 1.75 GB memory per VM instance. For example, the S1 pricing tier supports auto scale feature up to 10 VM instances. Let's say based on the demand for your app and the scaling rules, you have 4 VMs configured. 1.75 GB memory is per VM instance. So this means you have 4 times 1.75 GM memory available for your apps.
* 50 GB Storage - If you want to, you can run multiple applications using a single App Service Plan, but the improtant point to keep in mind is, these hardware resources are shared by all the apps. So this means, each app that is deployed in an App Service Plan gets a slice of computing power, memory and storage.

Please note : As of this course recording, if you compare S1 and P1V2 pricing tiers, S1 provides 100 ACUs (Azure Compute Units), where as P1V2 provides 210 ACUs. So you can say P1V2 is twice as powerful as S1 from computer power standpoint. So the point, that I am trying to get accross is, Azure Compute Units provides a way of comparing compute (CPU) performance. In general, the more Azure Compute Units you have, the more compute power available for your apps running in that App Servic Plan.

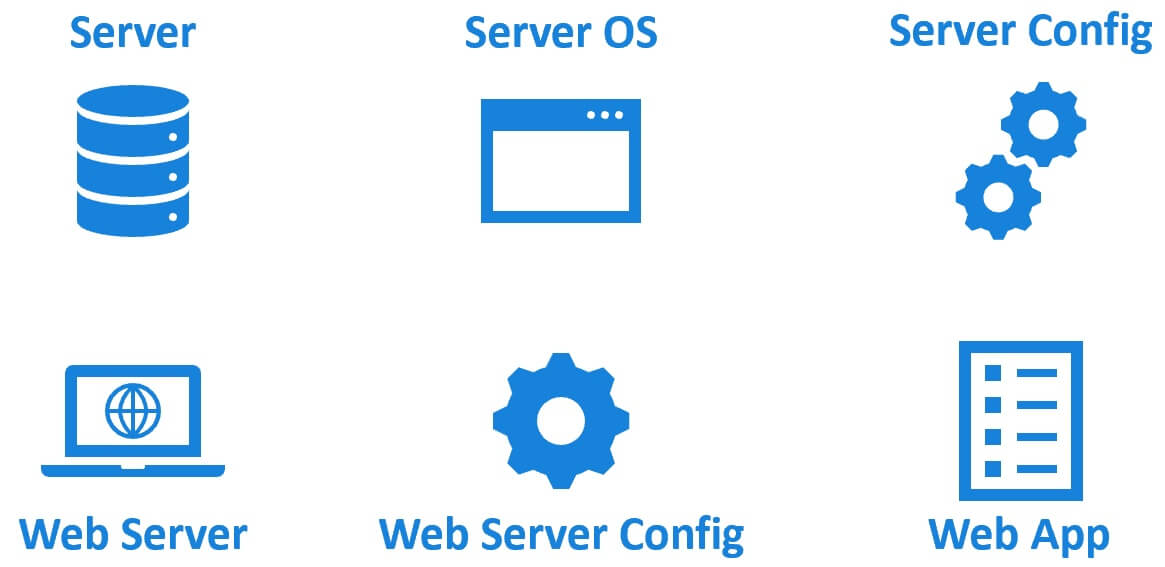
The monthly price you pay

Finally the monthly price you pay.

how to deploy a web application (specifically asp.net webforms application) in azure. It doesn't really matter if it's web forms, MVC, or .NET core, the process is the same. You will see how easy it is. With just a few clicks, in just a few minutes, you have your app up and running.

Hosting web application in your own on-premises server

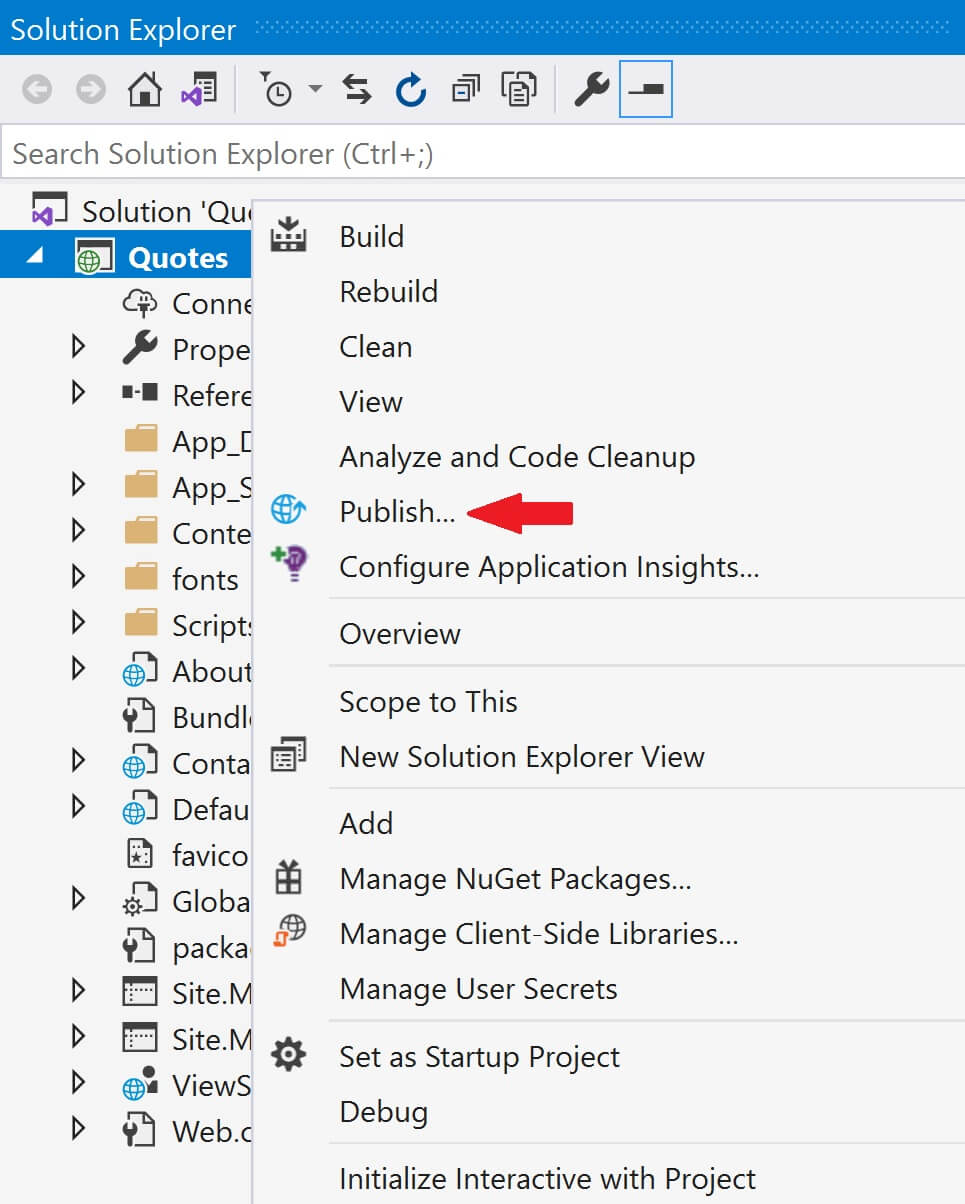
Compare this to hosting the application in your own on-premises server. It involves a lot of things.



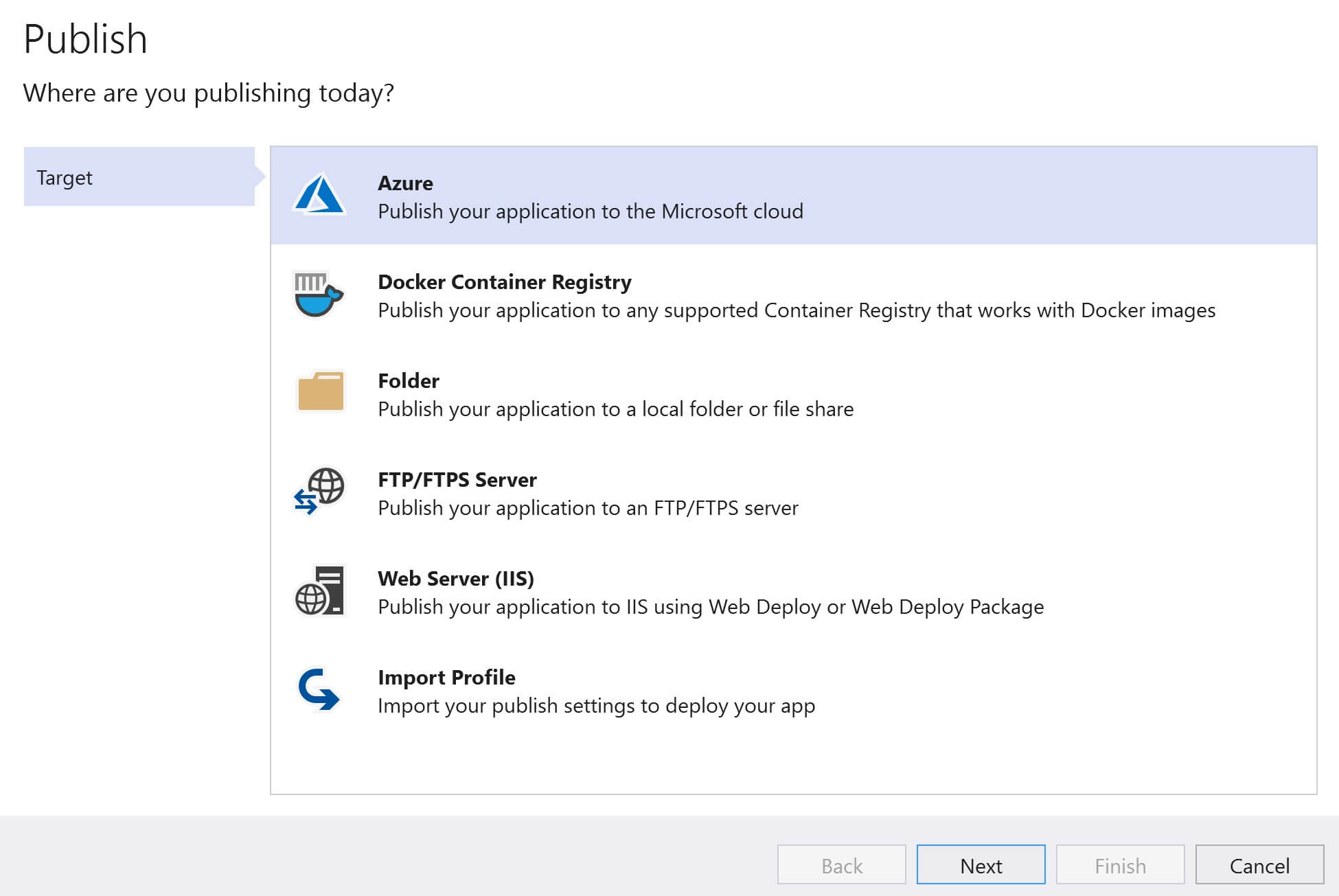
1. We need a physical or a virtual server
2. Server operating system like windows or linux must be installed
3. The server needs to be secured and configured. Install antivirus software, updates and patches. Configure firewall settings etc.
4. To host a web application and make it available on the web, you need a web server like IIS, Tomcat, NGINX etc.
5. You need to install and configure it correctly like security permissions, application pools, SSL setting etc.
6. On top of this, you install your web application and any dependencies it may have.

Deploying asp.net web application from visual studio to azure

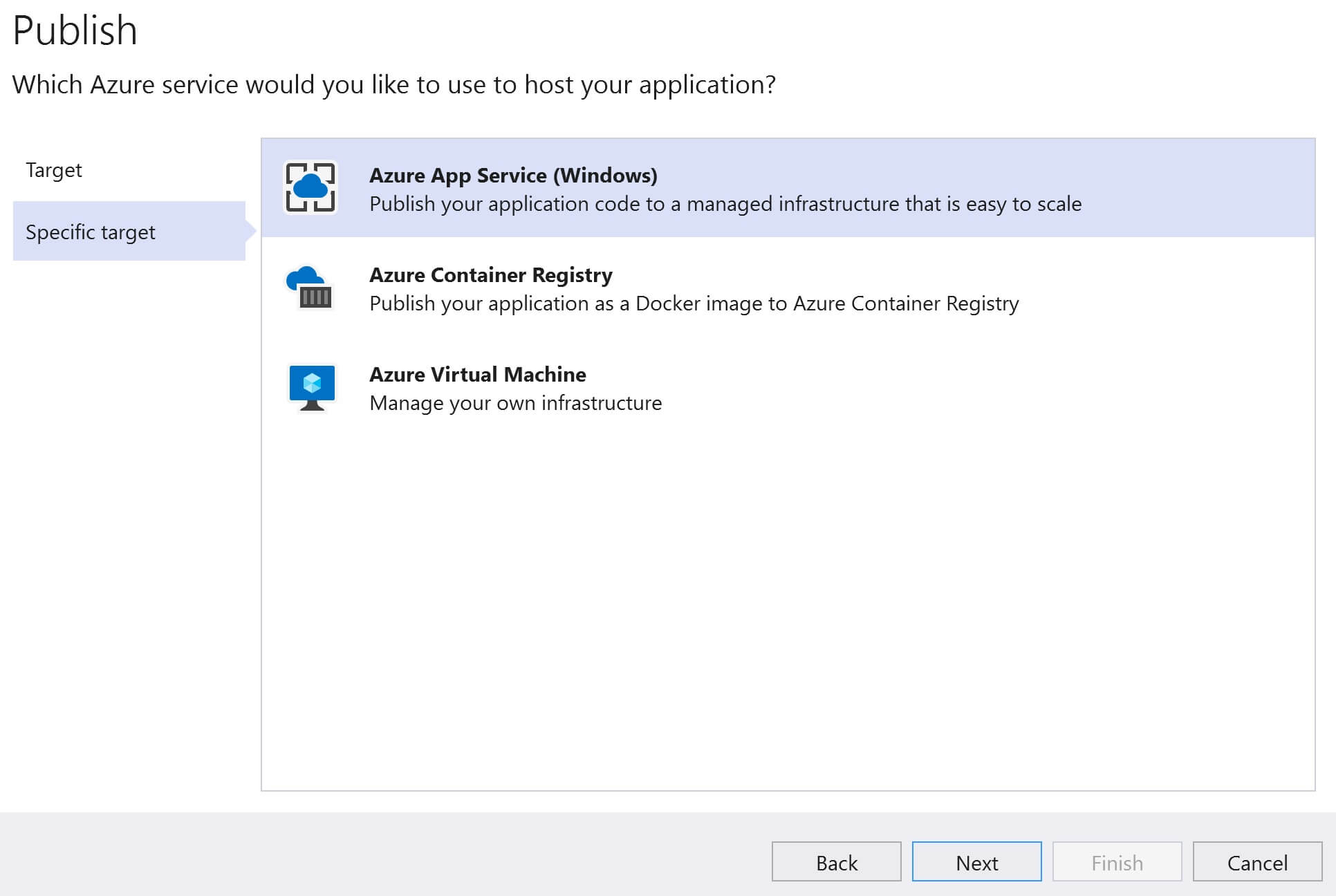
In the Solution Explorer, right click on the web application project and click on Publish



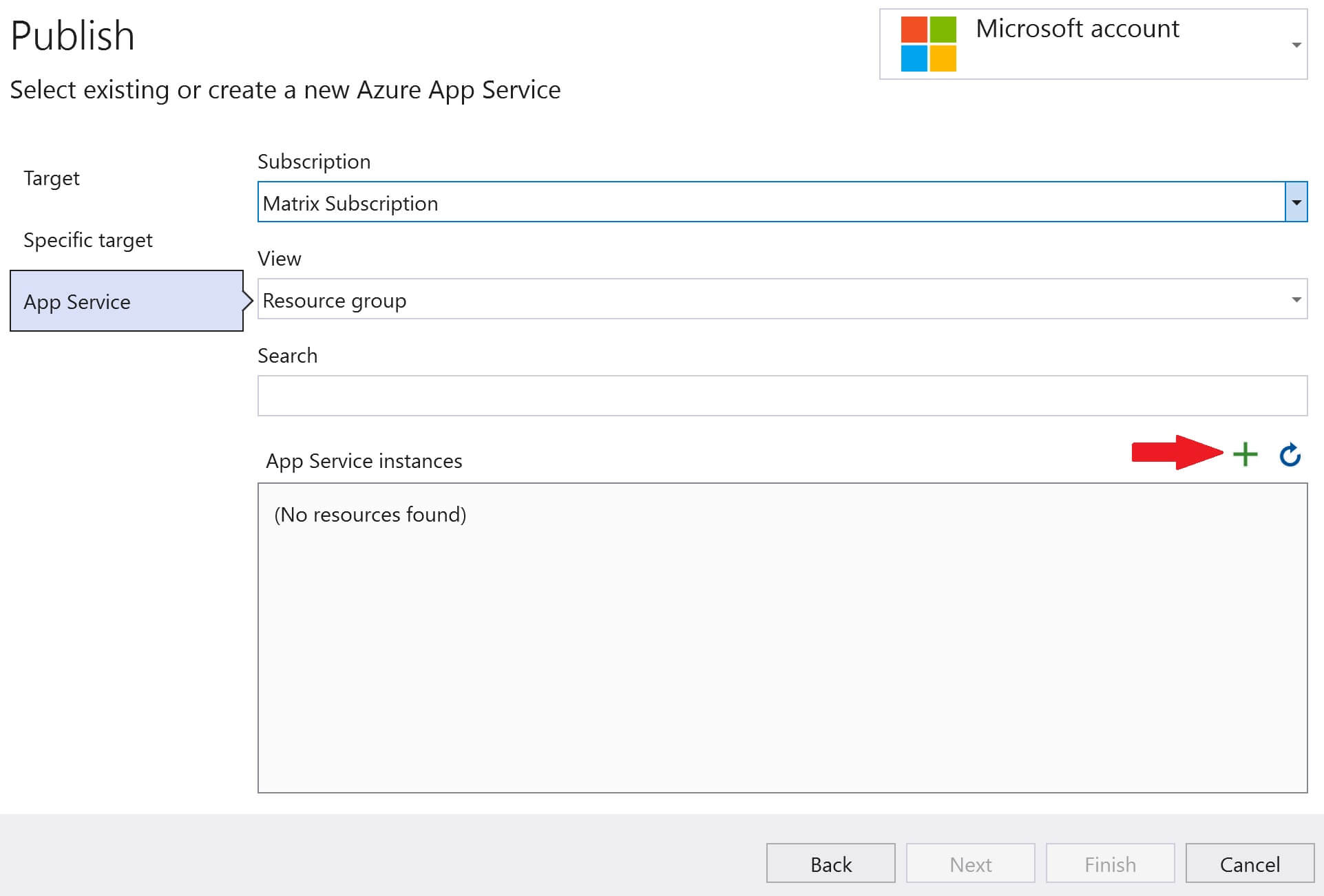
Select Azure and click Next



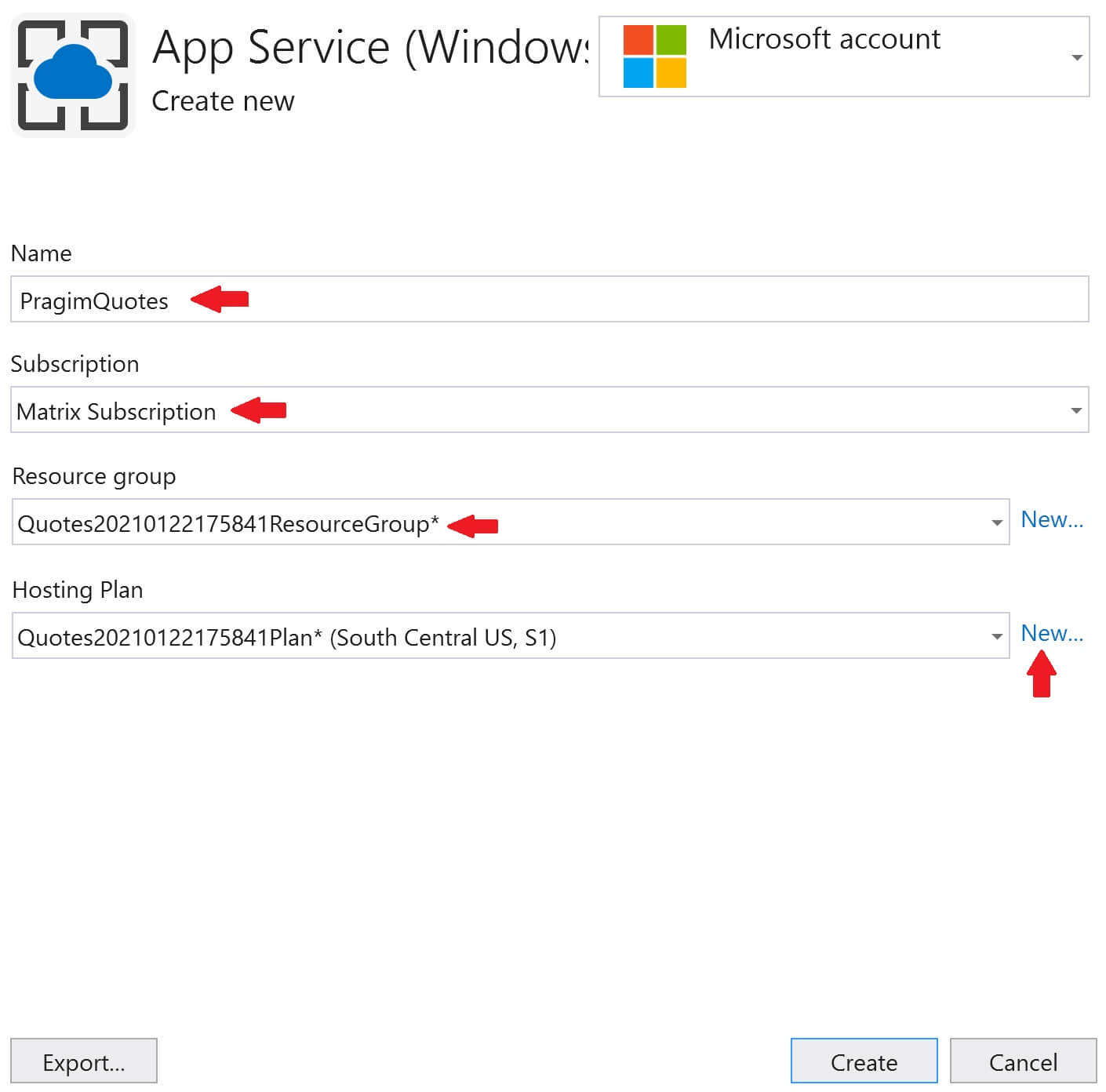
Select Azure App Service and click Next



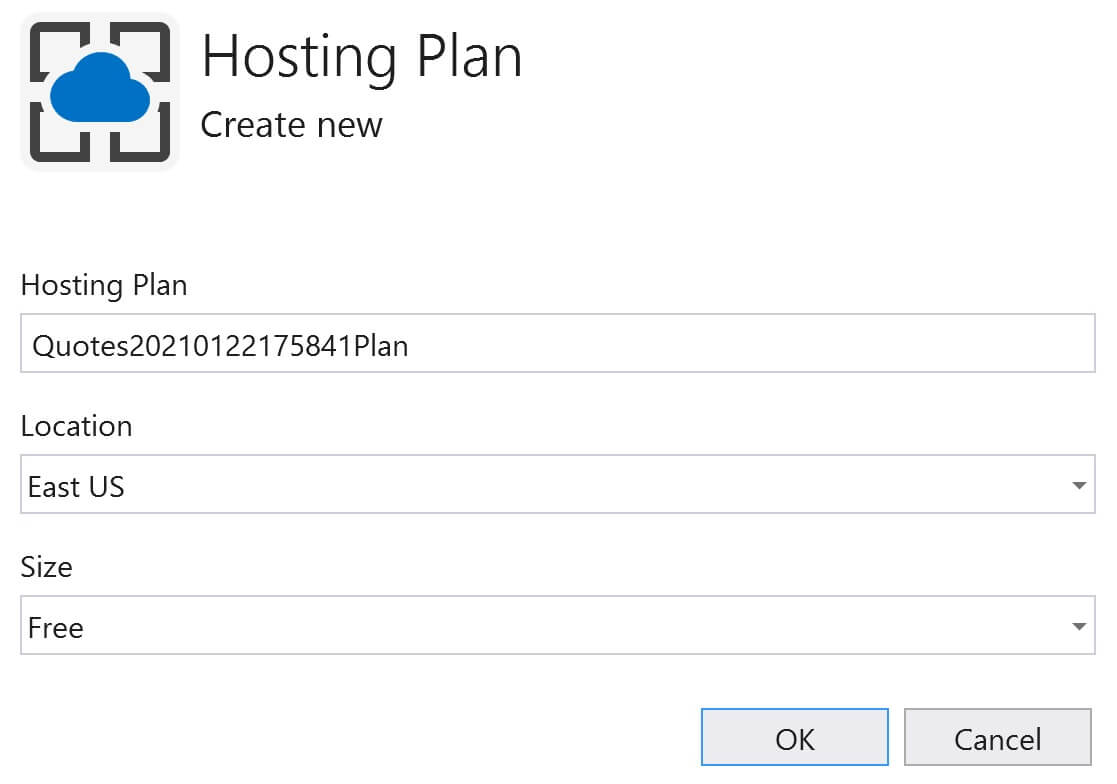
Click on the + sign



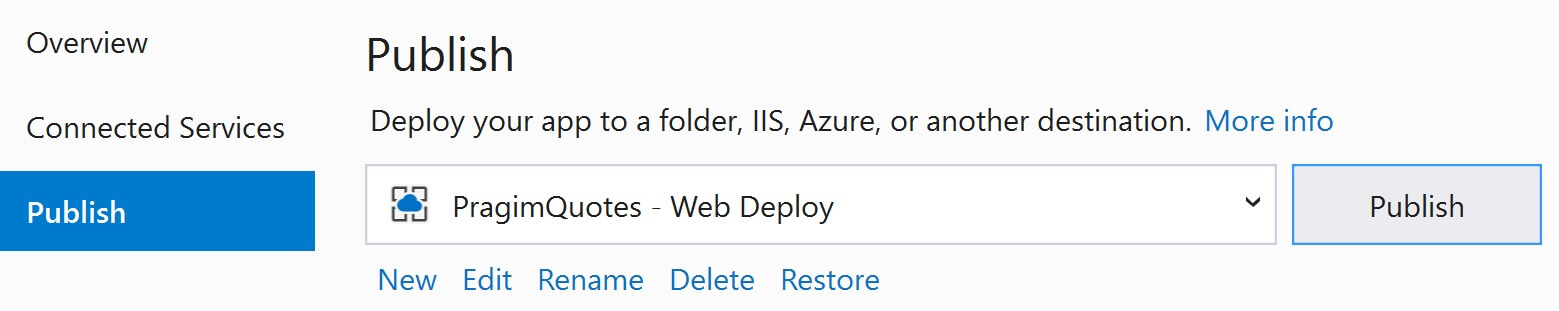
* Provide a Name for the App Service. This is the name that will be used to access the web application. firstDemo is the name I used, so to access the application I use the following URL in the browser.
* https://firstDemo.AzureWebSites.net
* Select your Subscription and Resource Group.
* Finally hosting plan. We want to use a free hosting plan. Click on the New link.



* Provide a name, location and the size for the Hosting Plan. It is the size that determines what you pay and what features you get. I selected the Free pricing tier.
* Remember to host a web application in Azure we need App Service and an App Service Plan. We discussed App Service and App Service Plan, including how to create them in the Azure portal, in our previous videos in this series.
* Right now, we are creating them (App Service and App Service Plan) from within visual studio instead of going through the azure portal.
* Click OK and the Finish.



You will then be back on the following Publish screen. In the dropdownlist, your App Service is selected. Click on the Publish button.



In a few minutes your app will deployed to azure and you can access it using the URL https://firstdemo.azurewebsites.net