Table of Contents

[AZURE COMPUTE SOLUTIONS 2](#_Toc133780149)

[DEPLOYING APP IN VM 2](#_Toc133780150)

[LOCAL PROJECT 2](#_Toc133780151)

[DEPLOYING THE WEB APP On WINDOWS VM 3](#_Toc133780152)

[ASSIGN/CONFIGURE DNS NAME TO VM 3](#_Toc133780153)

[ADDING INBOUND PORT RULE FOR PORT-8172 4](#_Toc133780154)

[ADD A ROLE OF MANAGEMENT SERVICE ON THE VM 4](#_Toc133780155)

[INSTALL THE .NET HOSTING BUNDLE 5](#_Toc133780156)

[INSTALL WEB DEPLOY TOOL (V3.6) 6](#_Toc133780157)

[PUBLISH THE LOCAL PROJECT FROM VISUAL STUDIO 6](#_Toc133780158)

[DEPLOYING THE WEB APP ON LINUX VM 8](#_Toc133780159)

[PUBLISHING THE APP LOCALLY 8](#_Toc133780160)

[PUBLISHING AN APPLICATION FROM GITHUB 9](#_Toc133780161)

[AZURE WEB APP 9](#_Toc133780162)

[PUBLISHING AN APPLICATION FROM VISUAL STUDIO TO AZURE WEB APP 10](#_Toc133780163)

[AZURE SQL DATABASE 10](#_Toc133780164)

# AZURE COMPUTE SOLUTIONS

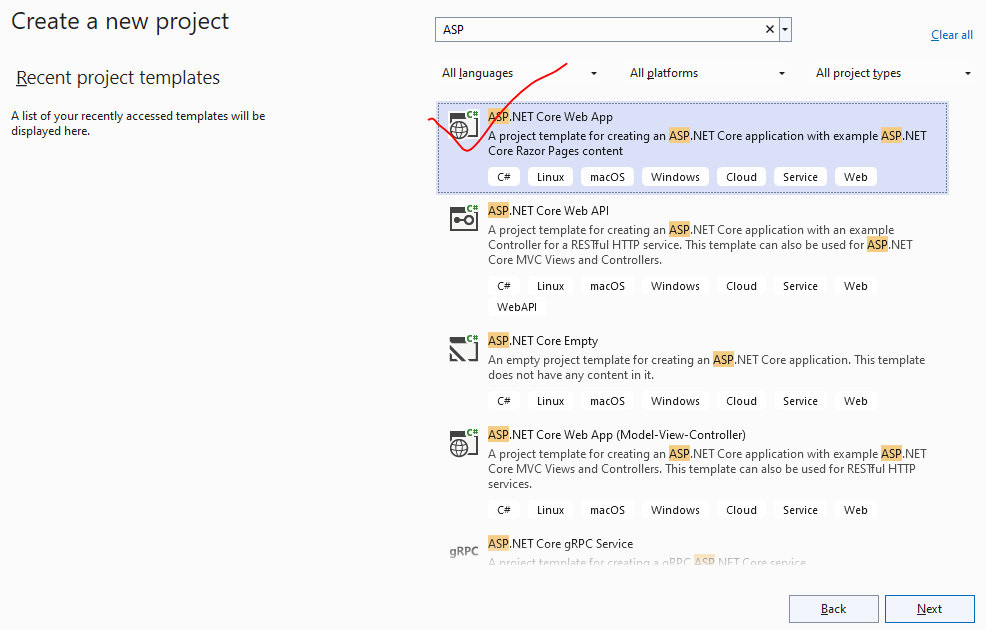
## DEPLOYING APP IN VM

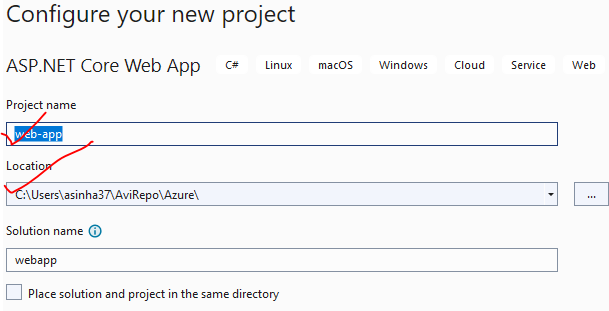
Steps to deploy the application

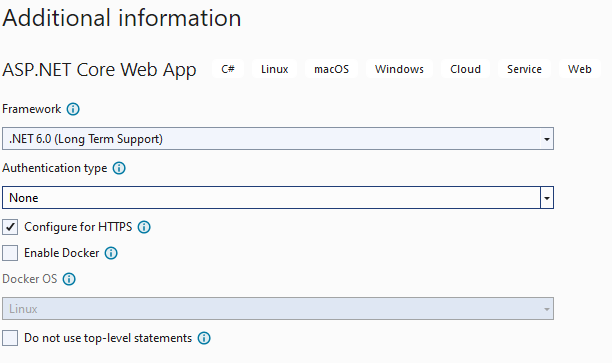
* Create a Project locally (.NET Application
* Provision the Window’s VM and Install IIS on VM.
  + PowerShell command to install IIS: **Install-WindowsFeature -Name Web-Server -IncludeManagementTools**
* Deploy the locally created project to IIS

## LOCAL PROJECT

* Open Visual Studio 🡪 New Project 🡪 Search for ASP .NET Core Project 🡪 Next
* After entering the require details (as shown below) 🡪 Create Project





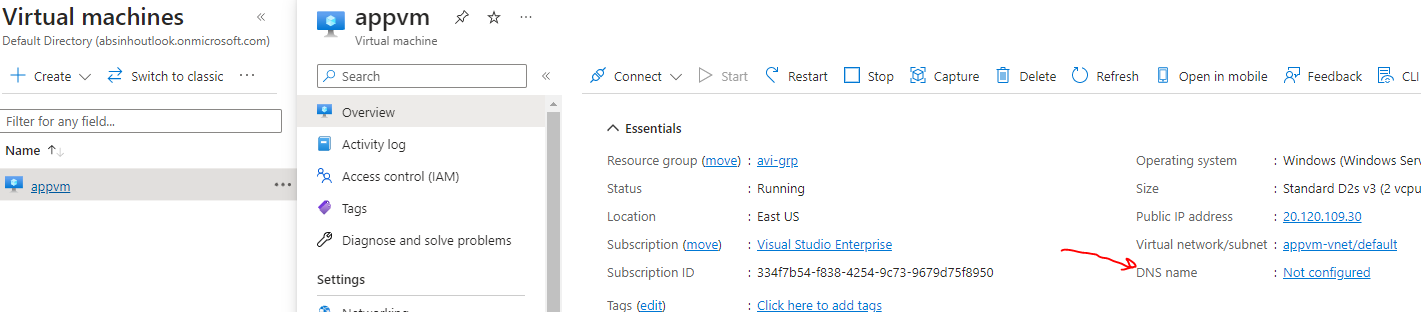


## DEPLOYING THE WEB APP On WINDOWS VM

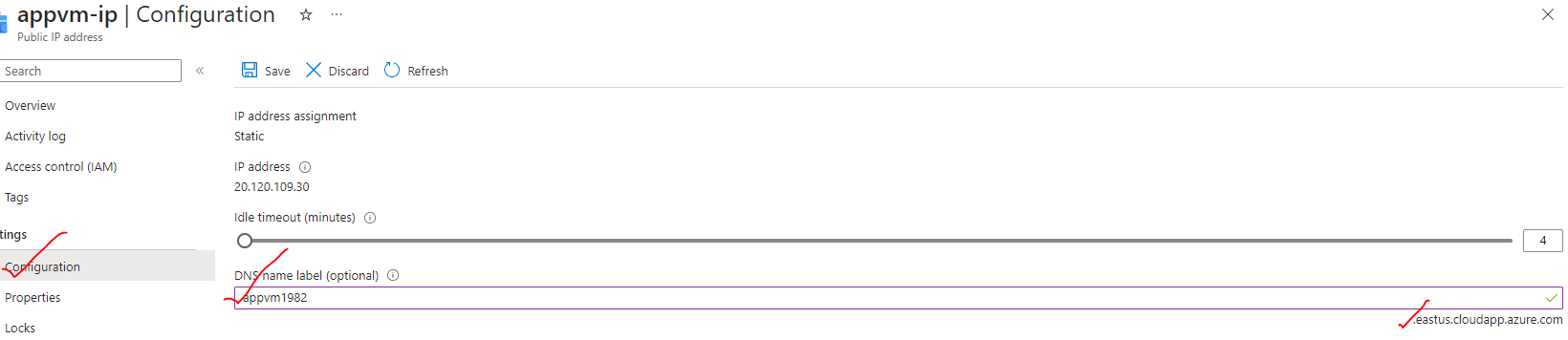
Step to deploy web application

1. **ASSIGN DNS NAME TO THE VM**
2. **ADD A RULE FOR PORT “8172” TO NSG**
3. **ADD A ROLE OF MANAGEMENT SERVICE ON THE VM**
4. **CHECK THE CONFIGURATION OF MANAGEMENT SERVICES IN IIS**
5. **INSTALL THE .NET HOSTING BUNDLE. THIS ALLOWS .NET APPLICATIONS TO BE HOSTED ON IIS**
6. **INSTALL WEB DEPLOY TOOL (V3.6)**

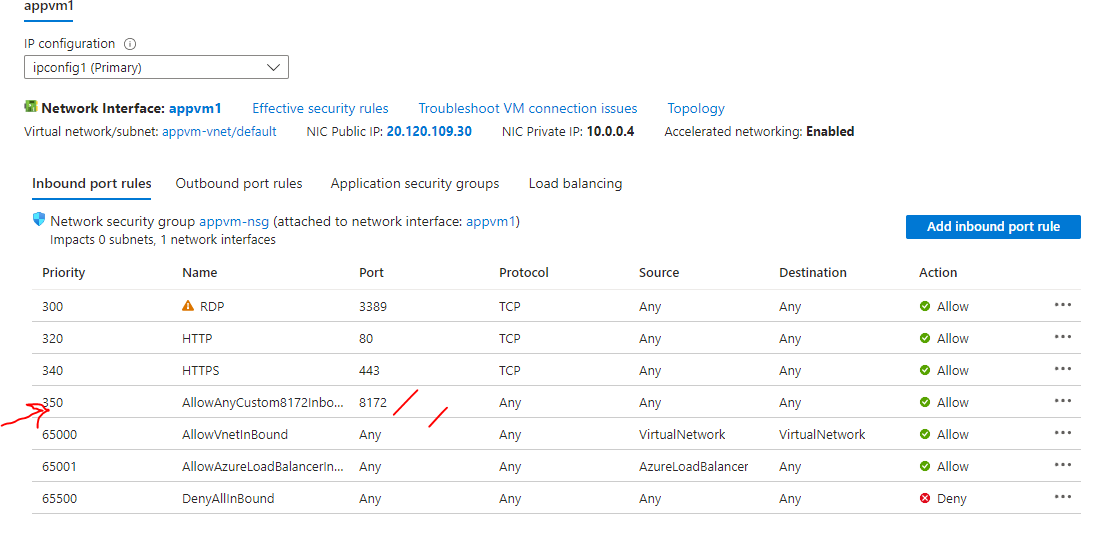
### ASSIGN/CONFIGURE DNS NAME TO VM



* Configure the DNS Name. Now the VM can also be access via DNS (Not just using public IP )
* Hence the VM can be accessed using - <http://appvm1982.eastus.cloudapp.azure.com/>

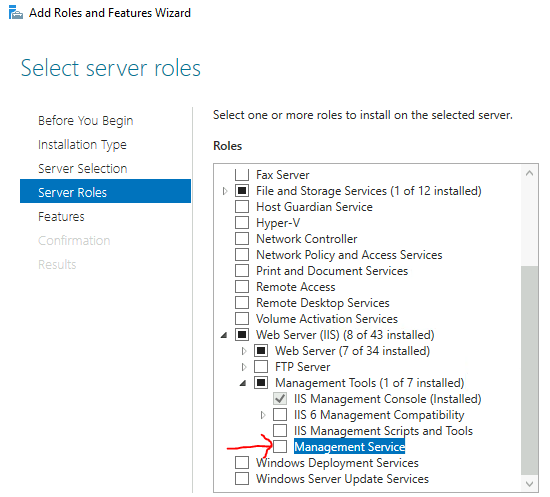


### ADDING INBOUND PORT RULE FOR PORT-8172

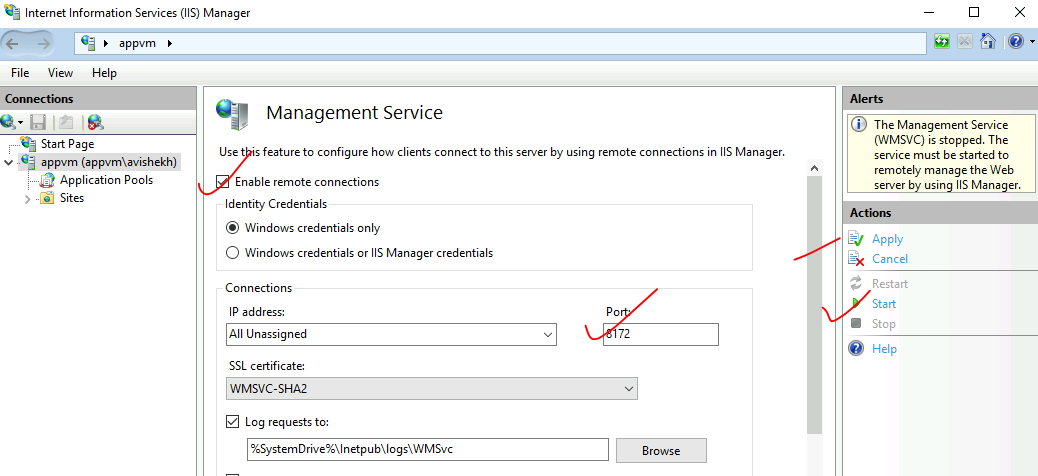
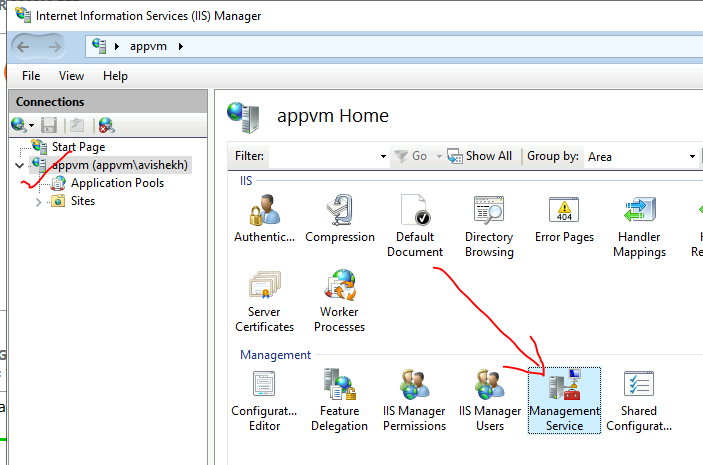


### ADD A ROLE OF MANAGEMENT SERVICE ON THE VM

* Go to Server Manager 🡪 Add Roles and Features 🡪 Select Management Service 🡪 Add Feature

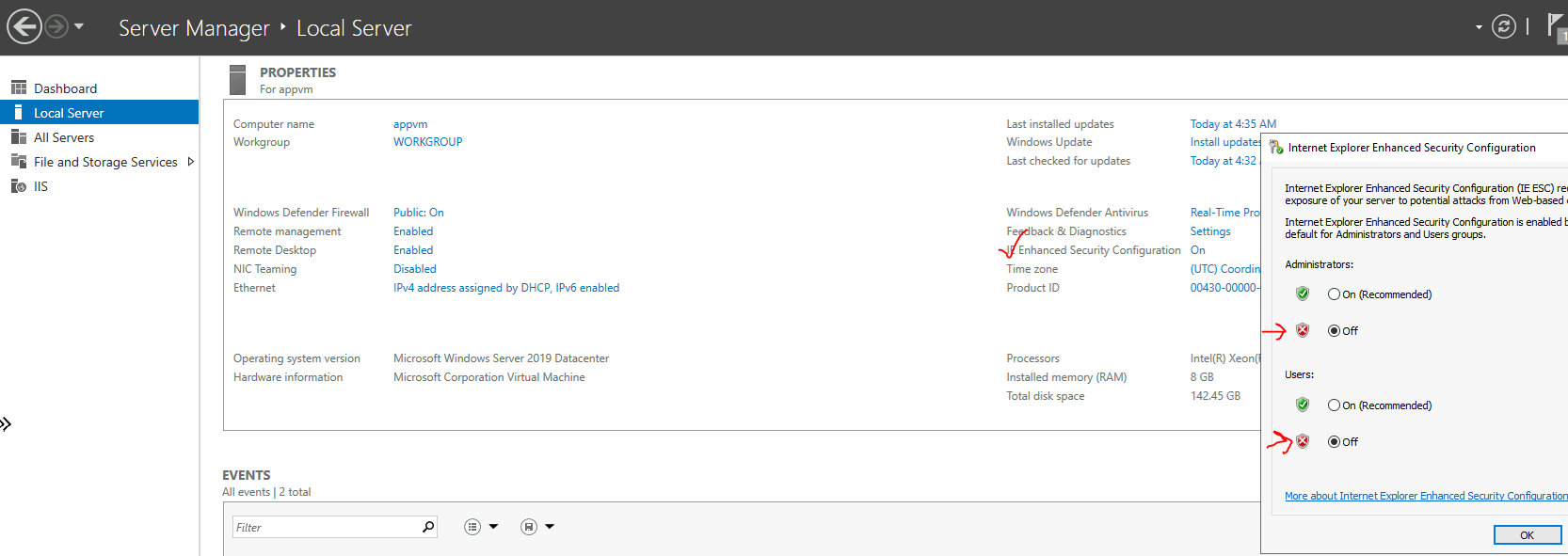


|  |  |
| --- | --- |
|  | * Open IIS from Server Manager Tools * Select the VM 🡪 Management Service🡪 Click to open it. * Select “Enable remote connections” * Apply and start the service. |

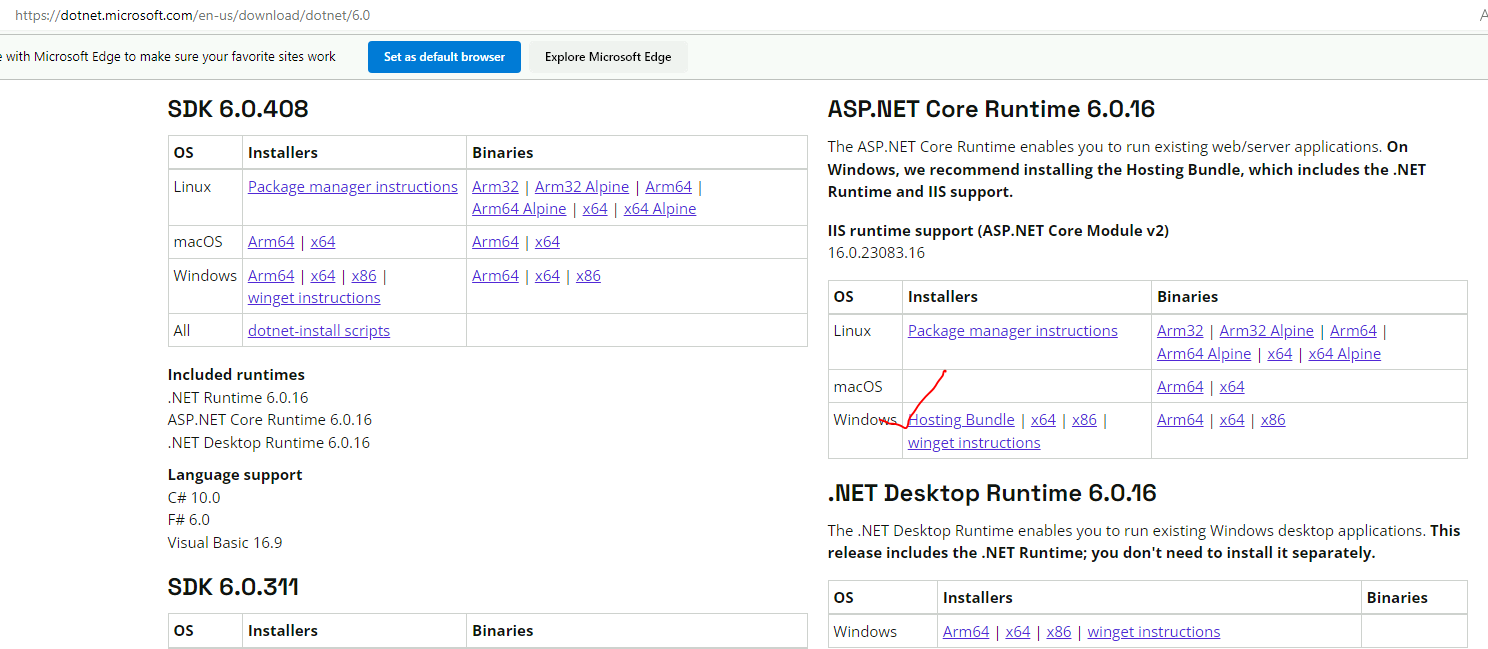


### INSTALL THE .NET HOSTING BUNDLE

* This is to install the .Net framework on the Hosting Platform(VM).
* In Server Manager 🡪 Local Server 🡪 IE enhanced security feature 🡪 Turn it off. This will allow us to download the .NET framework on the VM



* Download and install .NET platform. This installation is needed on the Hosting Platform(VM) to deploy any .NET based application.



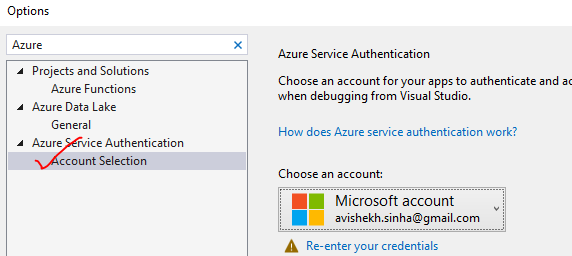
### INSTALL WEB DEPLOY TOOL (V3.6)

* This helps us in deploying / publish the application from Visual Studio. We have to do complete installation of web deploy tool.

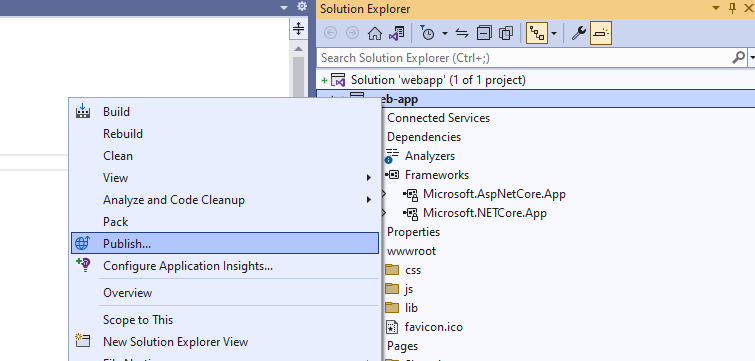
|  |  |
| --- | --- |
|  |  |

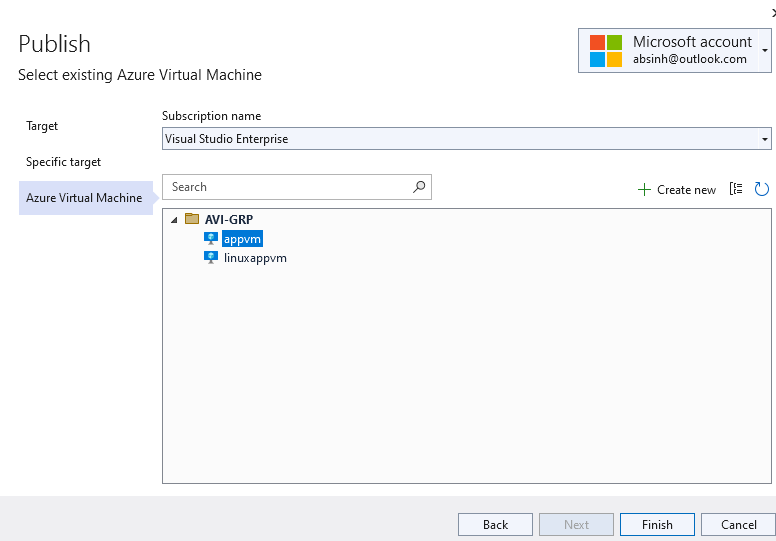
### PUBLISH THE LOCAL PROJECT FROM VISUAL STUDIO

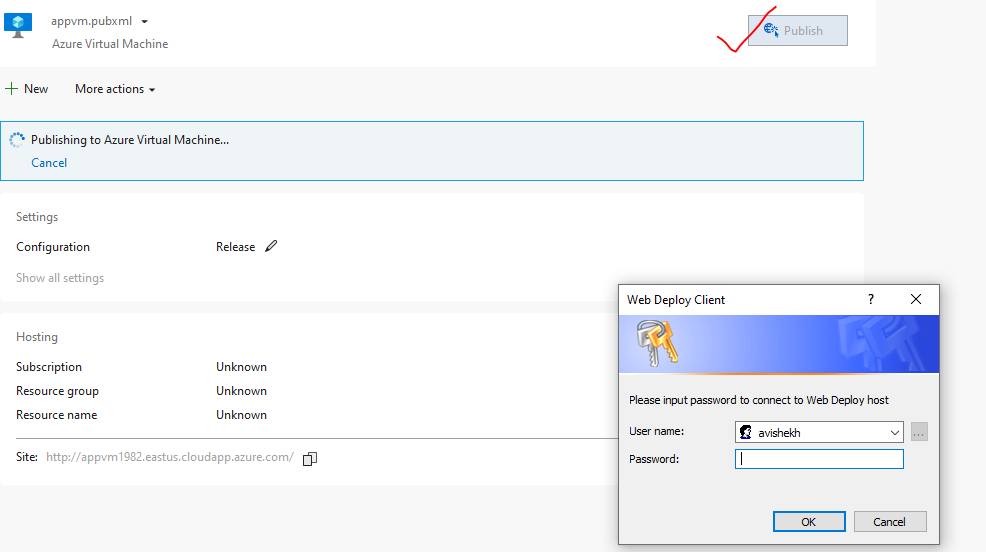
* Connect Azure with Visual Studio . Tools 🡪 Options



#### PUBLISH THE PROJECT







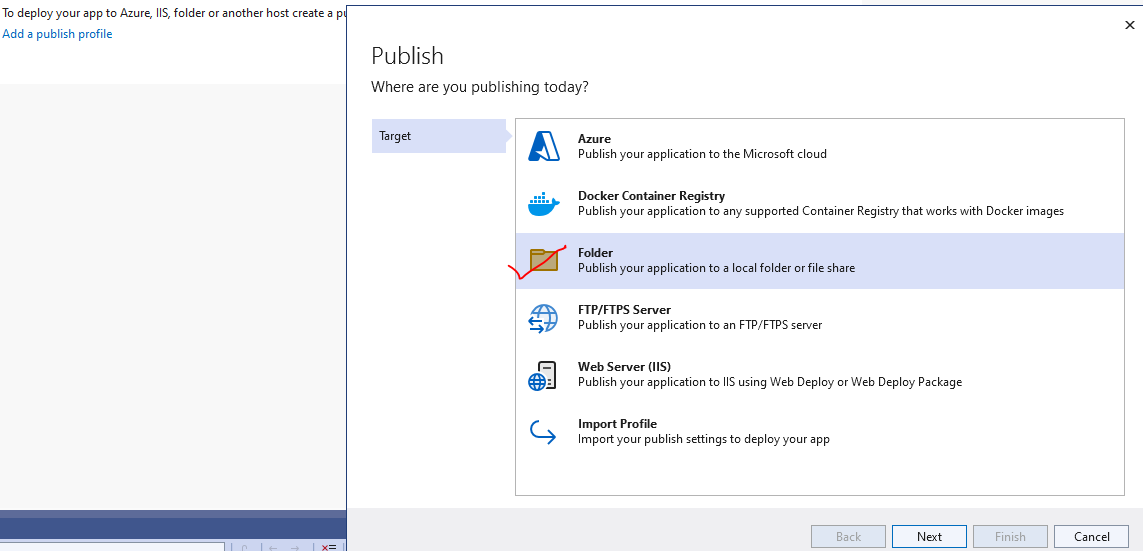
## DEPLOYING THE WEB APP ON LINUX VM

|  |  |
| --- | --- |
| **LOGIN TO LINUX VM** | ssh username@publicip |
| **UPDATING PACKAGE** | sudo apt update |
| **INSTALL .NET RUNTIME ON LINUX VM** |  |
| **INSTALLING PACKAGE(nginx)** | sudo apt install nginx |

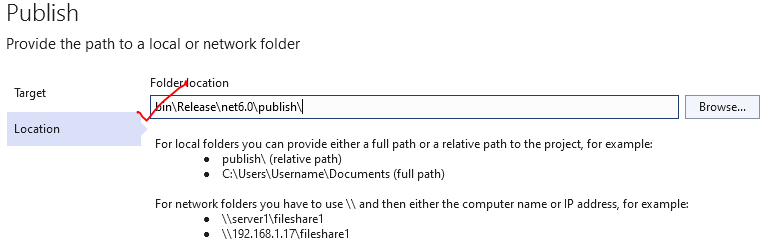
* In windows based VM we can be able to publish the project directly from Visual Studio. On the other hand – for linux VM we need to copy over the published file to hosting VM manually using WinSCP.
* The application must be published locally before copying.

### PUBLISHING THE APP LOCALLY

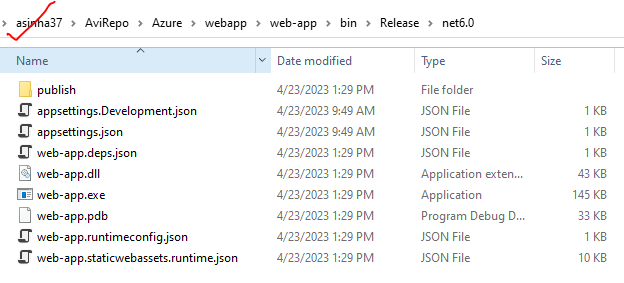
* Right click on the application in visual studio 🡪 Publish
* Create a new publish profile 🡪 Select Folder



* This is the path of local folder where the project will be published locally.



**LOCALLY PUBLISHED FILES**



wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb

sudo dpkg -i packages-microsoft-prod.deb

rm packages-microsoft-prod.deb

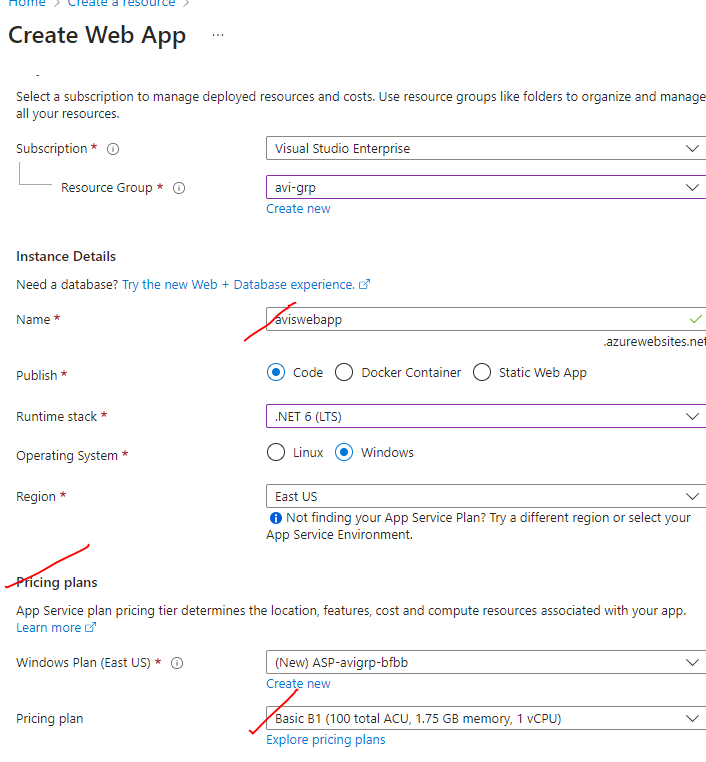
sudo apt-get update && \

sudo apt-get install -y dotnet-sdk-6.0

## PUBLISHING AN APPLICATION FROM GITHUB

# AZURE WEB APP

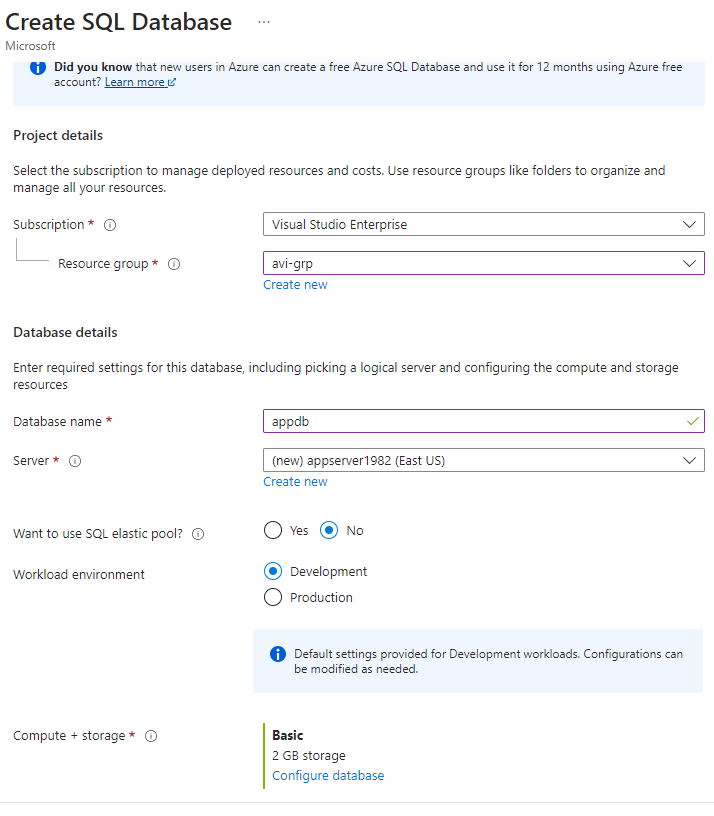
* It’s a PaaS service. The underlying VM also is managed by the platform itself. So instead of installing the .NET framework and installing IIS for hosting the web application - everything is available with the help of the Azure app service.
* The Azure app service also has other features such as
  + Auto scaling
  + Security and DevOps capability.
  + Support varieties of runtimes.
* When we create an Azure app, we also need to create an instance of an **app service plan**.
* The app service plan is what provides the underlying compute infrastructure for hosting the Azure web apps. App Service plan comes with different features and pricing models, depending upon the tier that we choose, it will allocate certain features for the underlying Azure web apps.

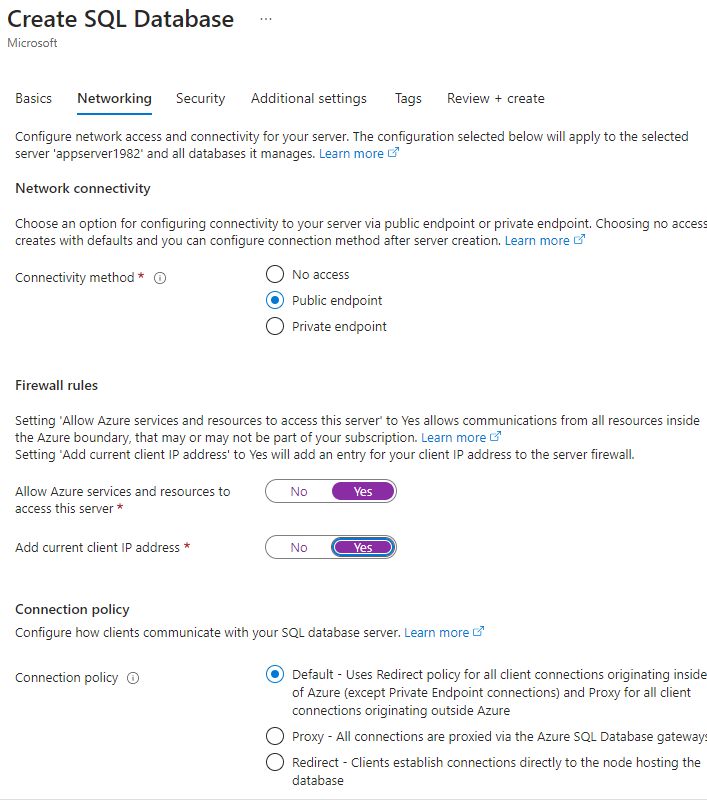


## PUBLISHING AN APPLICATION FROM VISUAL STUDIO TO AZURE WEB APP

# AZURE SQL DATABASE

* It’s a PaaS service.
* When we configure a Azure Database Service – it actually creates 2 resources
  + **THE SQL DATABASE**
  + **SQL SERVER**





1. Connect to the database with Query Editor and run the following query

|  |
| --- |
| CREATE TABLE Products  (  ProductID int,  ProductName varchar(1000),  Quantity int  )  INSERT INTO Products(ProductID,ProductName,Quantity) VALUES (1,'Mobile',100)  INSERT INTO Products(ProductID,ProductName,Quantity) VALUES (2,'Laptop',200)  INSERT INTO Products(ProductID,ProductName,Quantity) VALUES (3,'Tabs',300)  SELECT \* FROM Products |