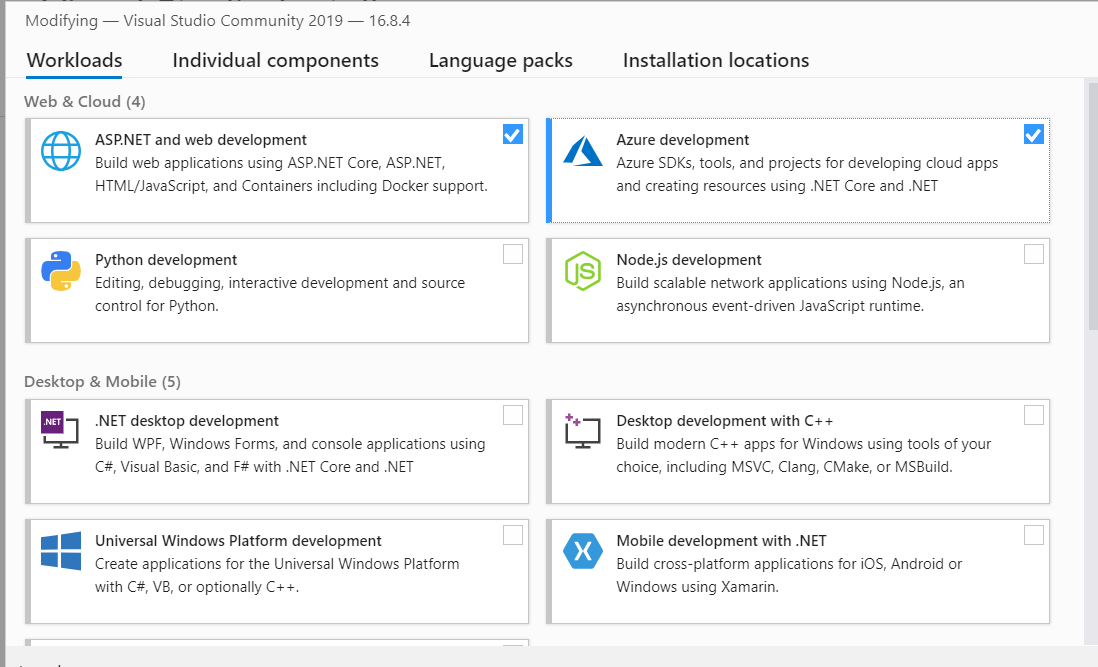
**Deploying the Application**:

At this point, we will have the Infrastructure built and configured to support deploying the application. Now it’s time to build the web application and get it deployed onto the VMs.

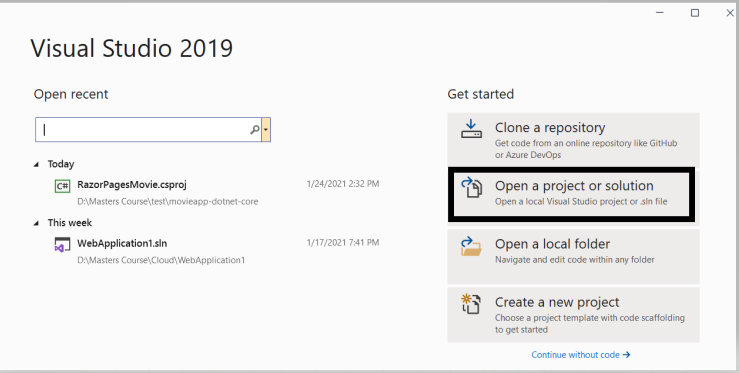
1. Download Visual-Studio Community Edition 2019.
2. Login to the Visual Studio using university email id.
3. Download the workloads ASP.NET and web development, Azure development, .Net Core cross-platform development.



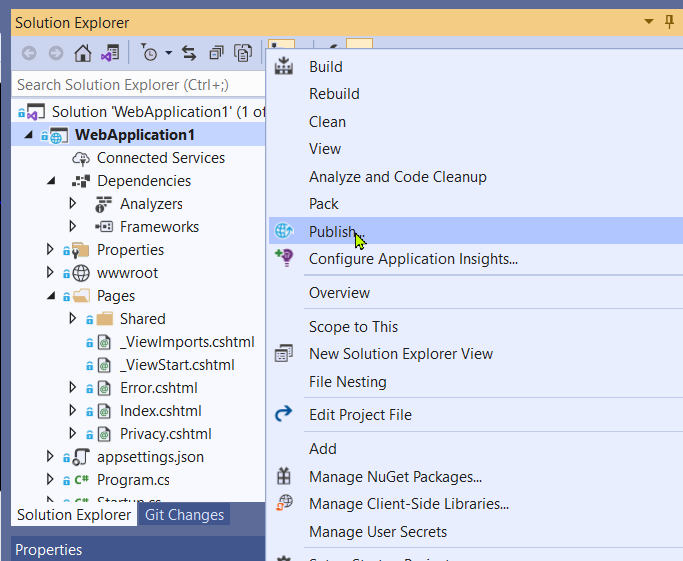
1. Once the required workloads are downloaded, we are ready to deploy an application to cloud.
2. Clone a sample app in a folder on the local machine. Below is the command to use the Git command-line.

Git clone https://github.com/Shubham-Girdhar/Infrastructure-As-Code.git

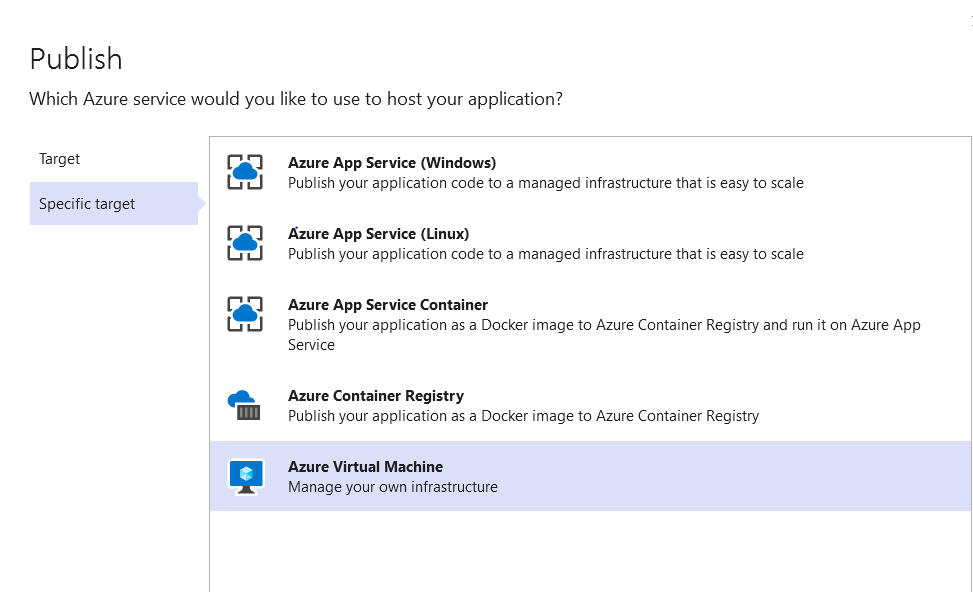
1. Open Visual Studio and click on “Open a project or solution”.



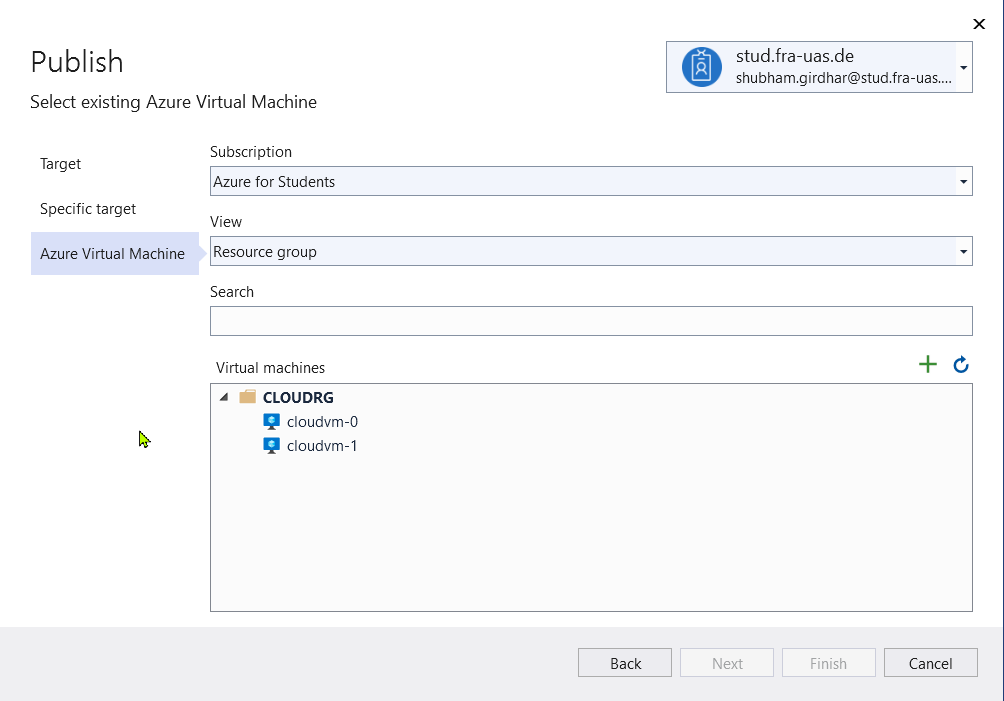
1. Open the folder where the application is cloned. Open the web application1 folder and select .csproj file or .sh file (These files specify a C# project file that contains a list of files that are included in the project to make the application work, along with the references to system assemblies).
2. Once the project is opened in Visual Studio, right click on the project and select **Publish** option.



1. This will open the publish window. Click on Azure and then on Specific target select **Azure Virtual Machine**.



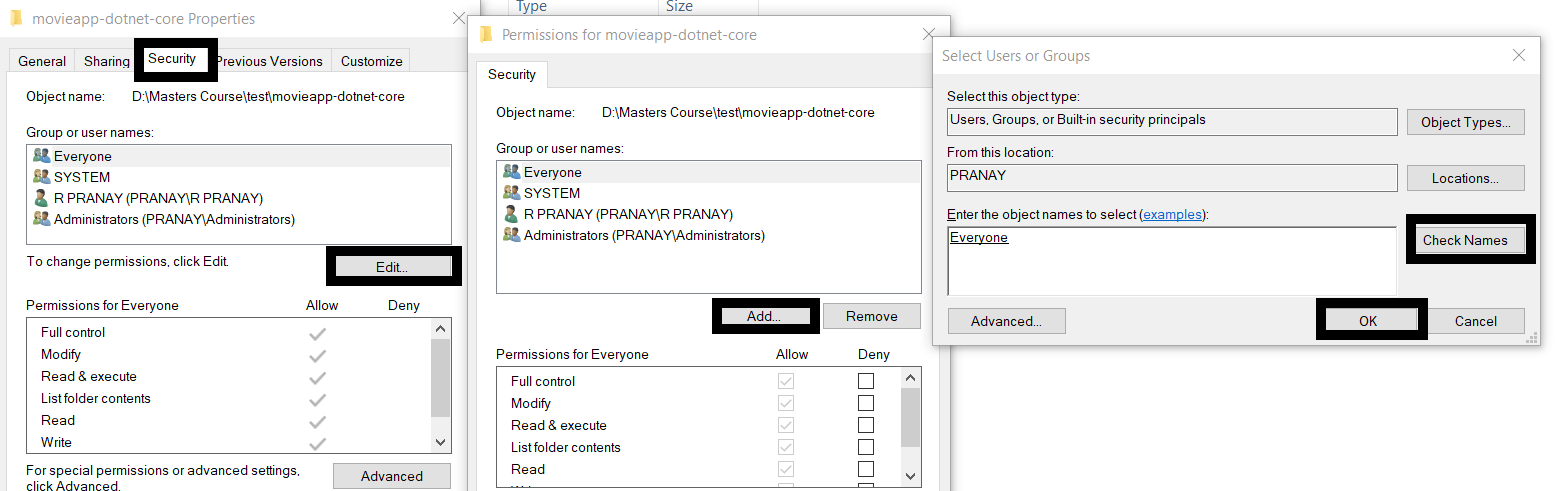
1. Once we select Azure Virtual Machine this will automatically fetch the details from our Azure cloud such as subscription type, resource group and also the available Virtual Machines (VMs).



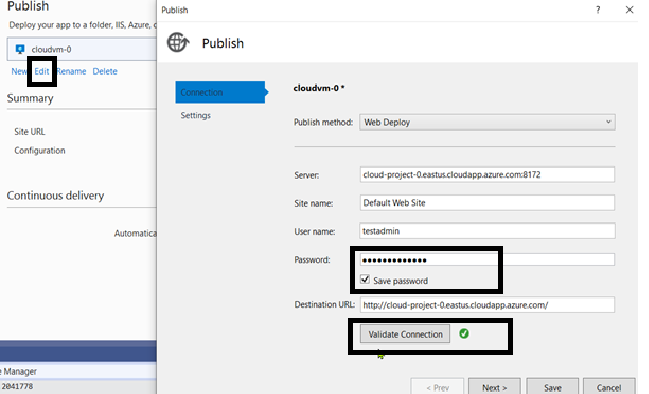
1. Select the first VM. We should see a progress circle below the window indicating that its testing the connection. If everything goes well, there will be no error messages. If so, click on **Finish** to add the VM.

**Note**: As we have cloned a project from github and trying to deploy it we might face an error in the above step. The error says “Web deploy cannot be succeeded”. This might be because of permission issues. This error occurs if we have used different accounts for Visual Studio and the local machine. For example, if the local machine is logged in with the account [xyz@gmail.com](mailto:xyz@gmail.com) and Visual Studio is logged in with [xyz@stud.fra-uas.de](mailto:xyz@stud.fra-uas.de) then this error might occur. In order to overcome this error, we need to enable permissions for that particular folder in the local machine.

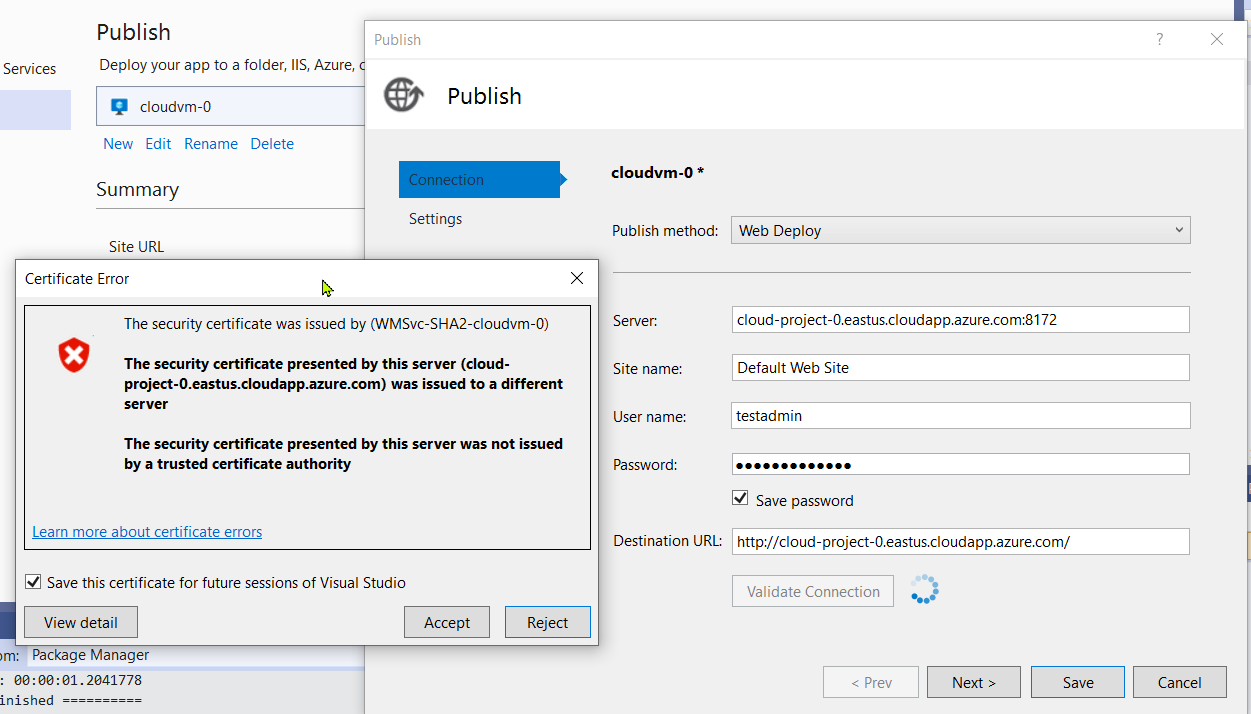
* Select the folder which contains the application and right click on it. Click on properties. Move to the security tab. Click on the Edit option, this will open the permissions of that folder. Click on Add, under Enter the object names to select type “Everyone” and select Check Names. Then click on OK and Apply. By performing the above process, we will enable the permissions to everyone and the error will be fixed.



1. Now, the next step is to provide the credentials to connect to the VM and also accept the self-signed certificate on it. To do that, click on **Edit** as shown below.



1. Provide the **Password** and click on **Save** **Password** so that there is no need to provide the password every time we perform a deployment. When complete, click on **Validate Connection.** The main reason for validating the connection is to permanently install the self-signed certificate on the VM. If not, we won’t be able to connect.
2. When we click on **Validate Connection** button, we will receive an error about not trusting a certificate. This is an expected event. Check the **Save this certificate for future sessions of Visual Studio** checkbox and click on **Accept.**



1. Now repeat the steps 6-13 for the second VM. Select the web application2 folder and run it on VM 2. Both the VMs have different web application deployed i.e., the structure of the application is same but the welcome message is different in each of the VMs. This is to know that which VM is picked by the load balancer in case of traffic.
2. Choose each VM and click on the **Publish** button. This action will tell Visual Studio to use WebDeploy to transfer all the files from local folders to the IIS site on each VM.
3. Once the **Publish** button is clicked a web browser will be opened with the deployed application.
4. Below are the images of the deployed web application in first and second VM respectively. For the same Load balancer IP we can see that same application is running in different VMs.

