**Application Deployment on Azure (PAAS)**

1. **IMPORTING THE DATABASE ON AZURE SQL SERVER**

* Using Data tier Application, the database is exported with schemas and all tables from SQL Server Management Studio to a storage account on Azure as a BACPAC file.
* After the BACPAC file was uploaded, the Azure SQL server was created, and the database was imported by selecting the BACPAC file in the storage account.
* The database is now imported to Azure SQL Server.

1. **DEPLOYING .NET CORE WEB API ON AZURE APP SERVICE**

* Before publishing the web API, the connection string in appsettings.json should be changed from the local server connection string to the connection string of the Azure SQL database .
* The web API from Visual Studio is published to the Azure App Service by right clicking over the project file and clicking the publish option. Then we can sign-in to our Azure account and create an Azure App Service and all the required configurations are done.
* So, after these steps, the NET CORE WEB API is deployed on Azure.
* We can access the API using the URL present on the Azure portal in the app service.

1. **DEPLOYING ANGULAR FRONTEND ON AZURE APP SERVICE**

* The first step, before deploying, is tochange the base API URL.
* Inside the SRC folder we will have the environment folder =>environments.prod.ts. The base API URL of our web API, which we deployed on Azure App Service, will be included in this file.
* Inside the environment.ts file, the base API of the local server is maintained, which will help run the app locally.
* After the configurations are done in the environment, the environment files should be imported to the services and changes should be made to the services accordingly.
* The Dist folder is now created using the command ng build. This is the folder used to deploy.
* Then in VS code, the Azure App Service extension is installed, and the Azure App Service is created on Azure.
* The dist folder is deployed on Azure. After the deployment is completed, the URL of the frontend should be added to the CORS of the web API so the frontend would be allowed to make requests.

Finally, we can use the front-end app service’s URL to access our application.