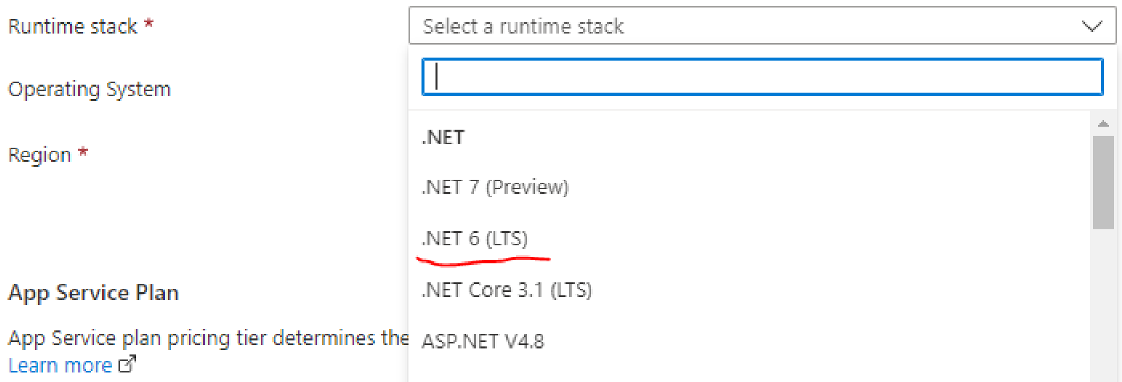
Deploying .netcore Web API, Angular UI and sql Database to Azure App Services

❶Creating Web app for Web API and angular in azure.

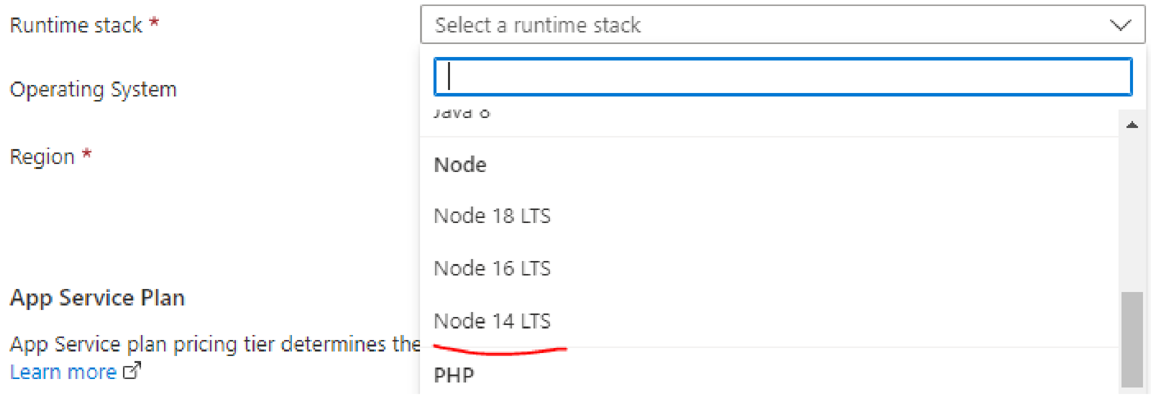
1. In azure portal , create azure app services resource for Web API and Angular.

2. While creating it for API choose run time stack as .NET 6 and while creating it for angular choose it as Node 14 or accordingly.

.net Web API

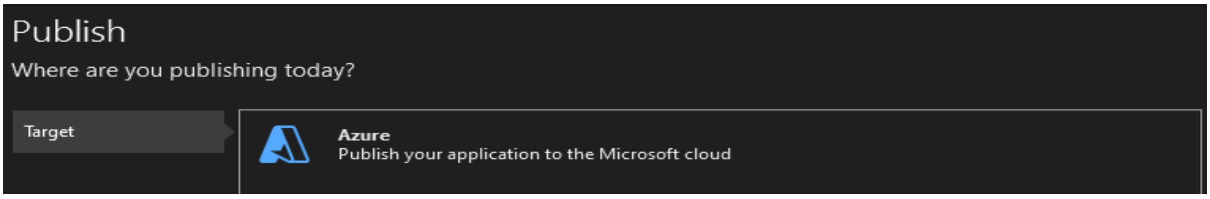


Angular

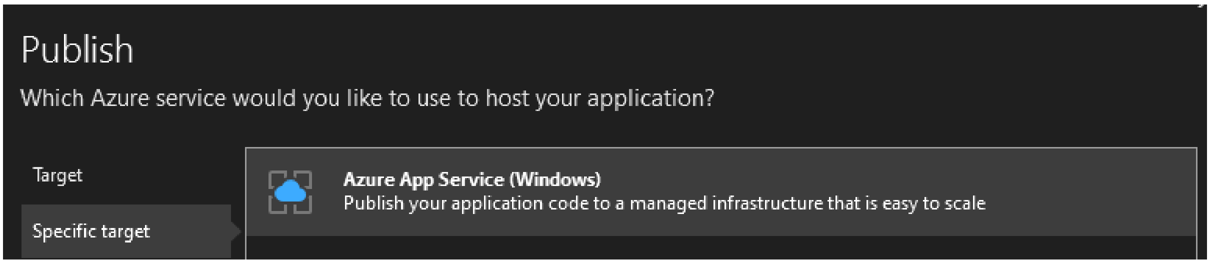


❷ Deploying Web API from Visual Studio 2022

1. Open your web Api Code in VS 2022.
2. Click On project file and say Publish.
3. Select Publish target as azure.


Publish
Where are you publishing today?
Target
Azure
Publish your application to the Microsoft cloud

4.Select Specific target as Azure App services.

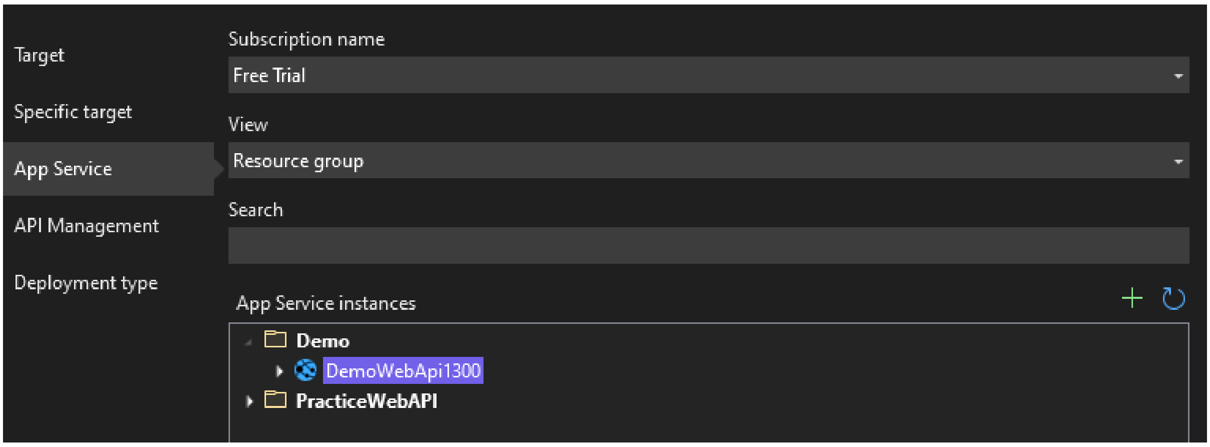


Publish

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

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

5. Select the web api app service that we created in first step .



Subscription name: Free Trial

View: Resource group

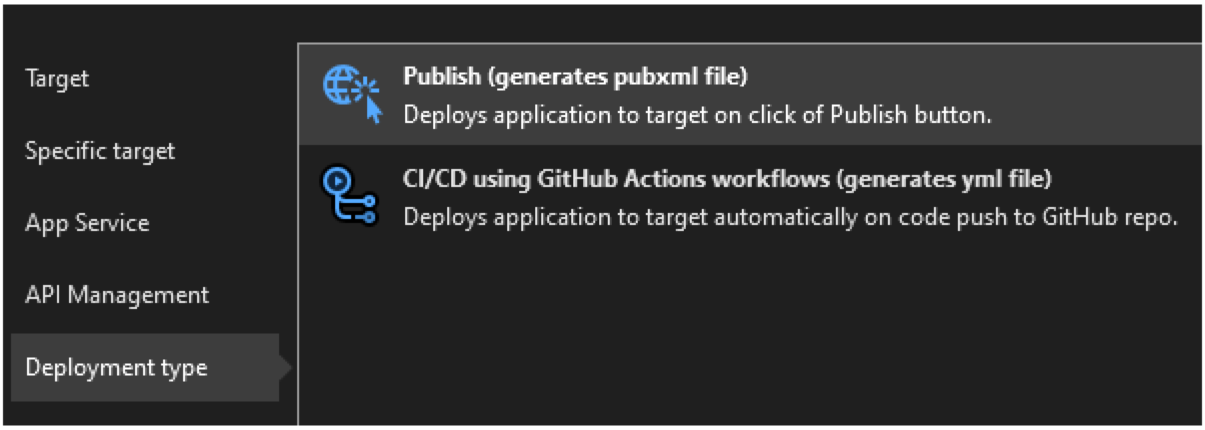
Search

App Service instances

- Demo
  - DemoWebApi1300
- PracticeWebAPI

6. Skip API Management.

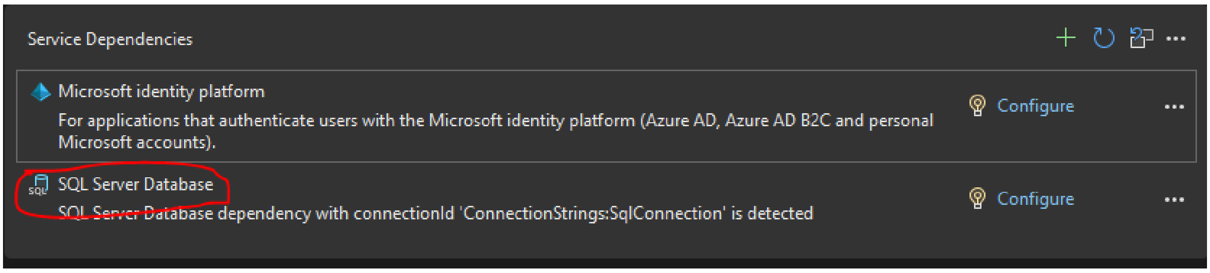
7. Select deployment type as publish, and finish.



Publish (generates pubxml file)
Deploys application to target on click of Publish button.

CI/CD using GitHub Actions workflows (generates yml file)
Deploys application to target automatically on code push to GitHub repo.

8. Configure Sql dependency

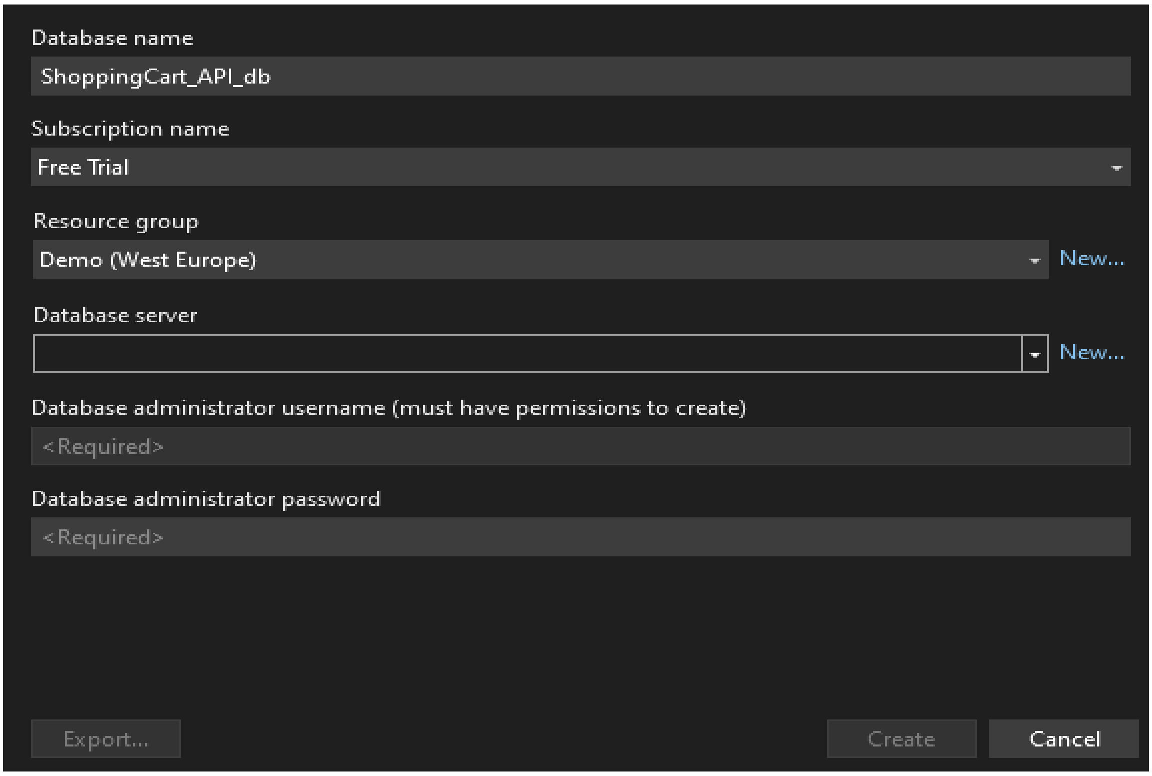


Service Dependencies

Microsoft identity platform
For applications that authenticate users with the Microsoft identity platform (Azure AD, Azure AD B2C and personal Microsoft accounts).

SQL Server Database
SQL Server Database dependency with connectionId 'ConnectionStrings:SqlConnection' is detected

9. Select service dependency as azure database services.
10. Create new database .



Database name: ShoppingCart_API_db

Subscription name: Free Trial

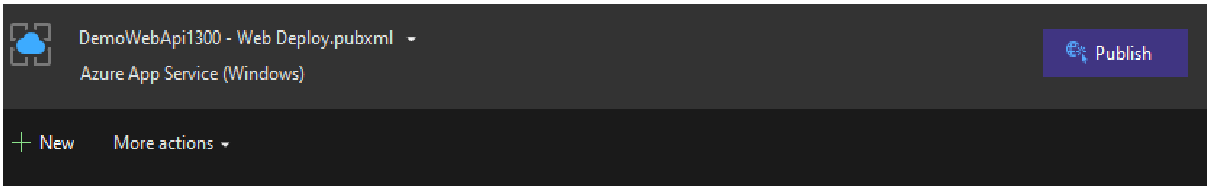
Resource group: Demo (West Europe)

Database server:

Database administrator username (must have permissions to create): <Required>

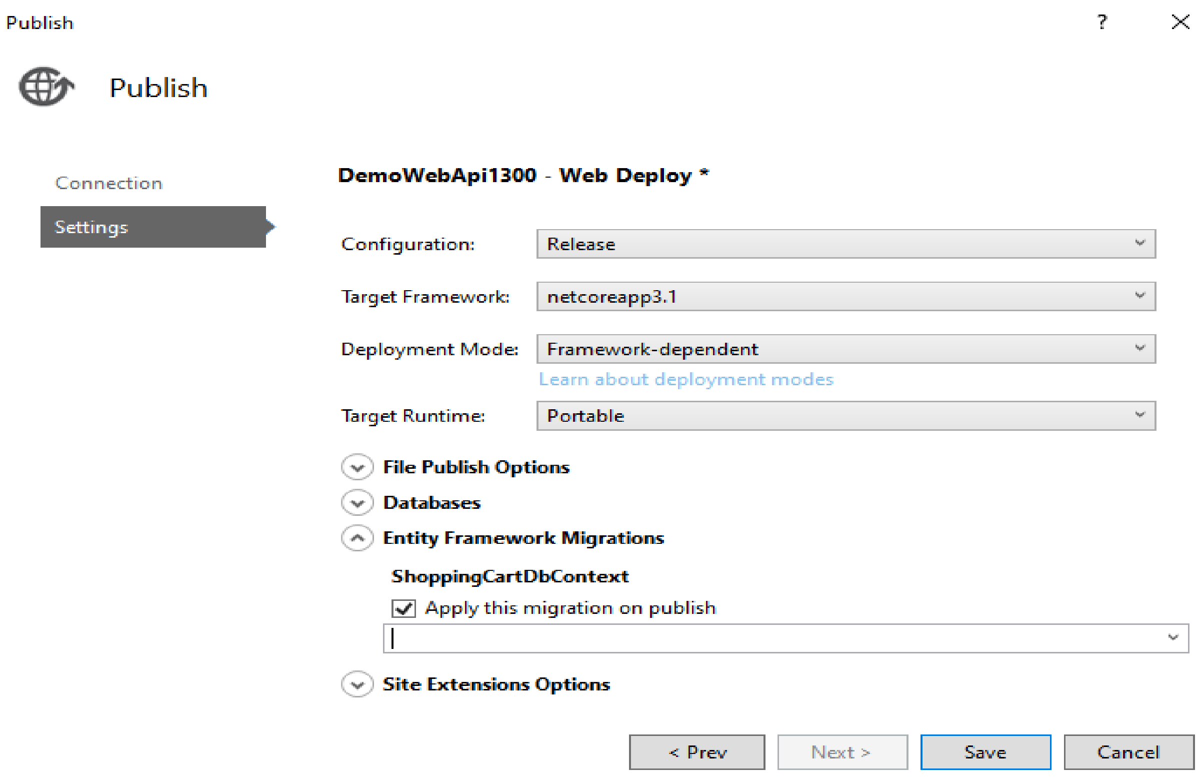
Database administrator password: <Required>

11.After configuration in more action, select edit and say yes to “Apply this migration to publish and save”



DemoWebApi1300 - Web Deploy.pubxml

Azure App Service (Windows)



DemoWebApi1300 - Web Deploy *

Configuration: Release

Target Framework: netcoreapp3.1

Deployment Mode: Framework-dependent

Learn about deployment modes

Target Runtime: Portable

File Publish Options

Databases

Entity Framework Migrations

ShoppingCartDbContext

☑ Apply this migration on publish

Site Extensions Options

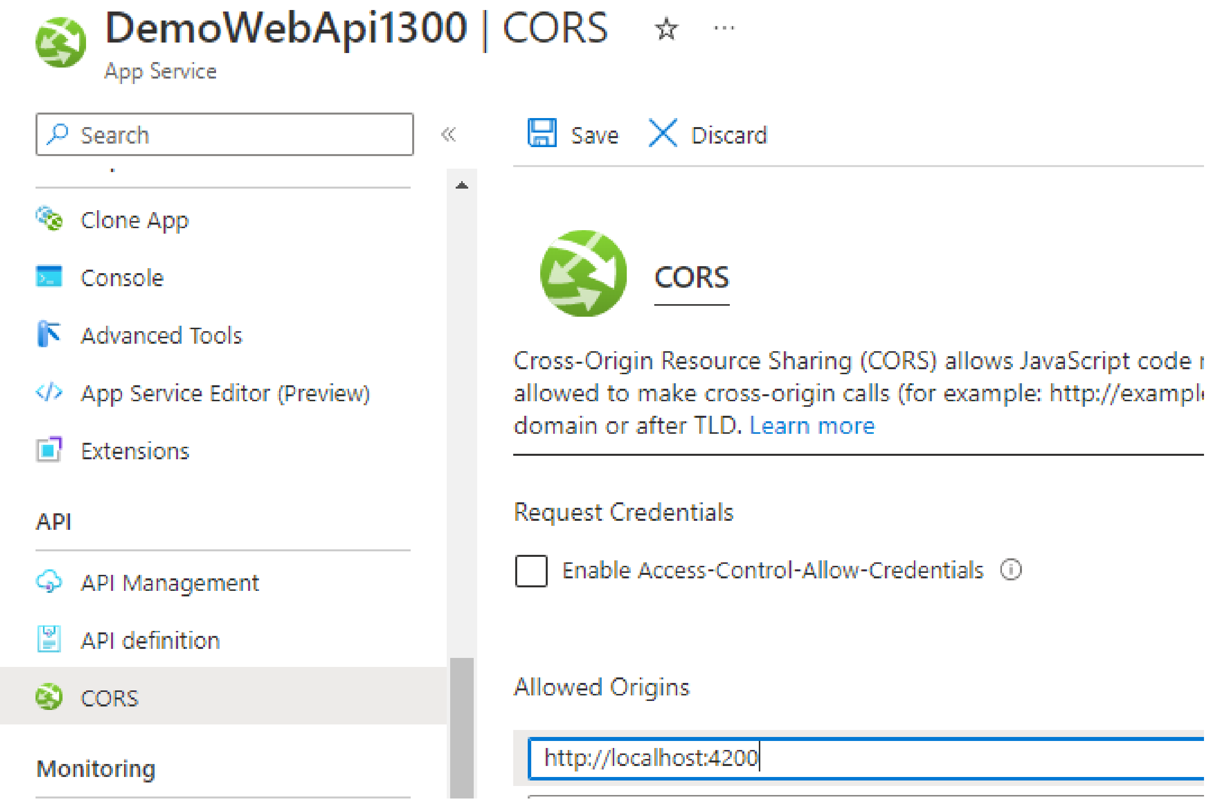
12. After this we are ready to publish !

❸ Deploying Angular Web App from Visual Studio Code

1. Open Angular frontend project in VS Code.

2. Paste URL of the web api wherever needed in VS code services.

3. Before trying out if connection is established , update CORS permission in the WEB API that we just deployed in azure.



Cross-Origin Resource Sharing (CORS) allows JavaScript code r
allowed to make cross-origin calls (for example: http://exampl
domain or after TLD. Learn more

Request Credentials

□ Enable Access-Control-Allow-Credentials

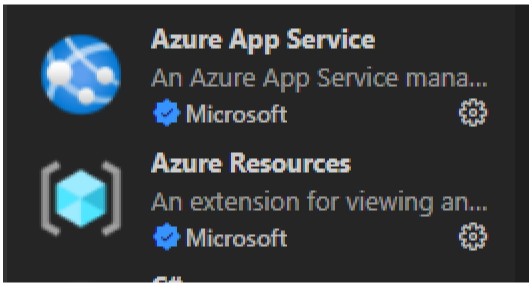
Allowed Origins

http://localhost:4200

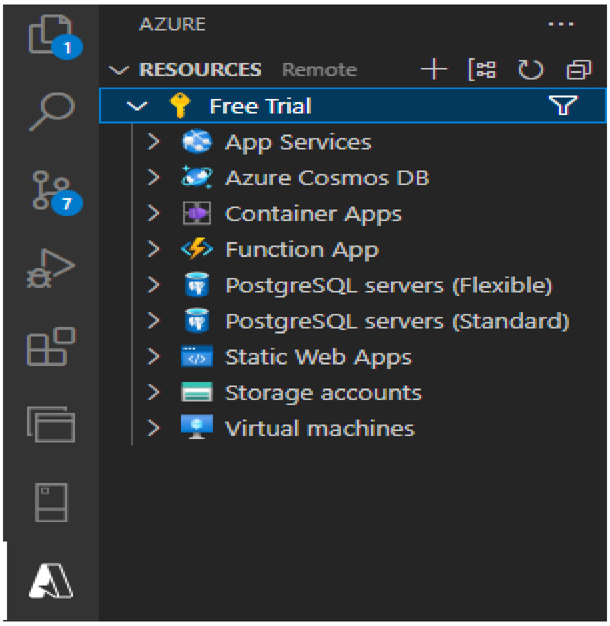
4. try running application using command “ng serve --configuration=production”

5. If it works , run command “ng build --configuration=production”.

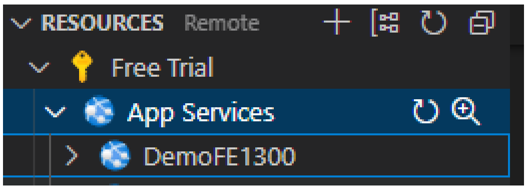
6. Install Extensions as follows



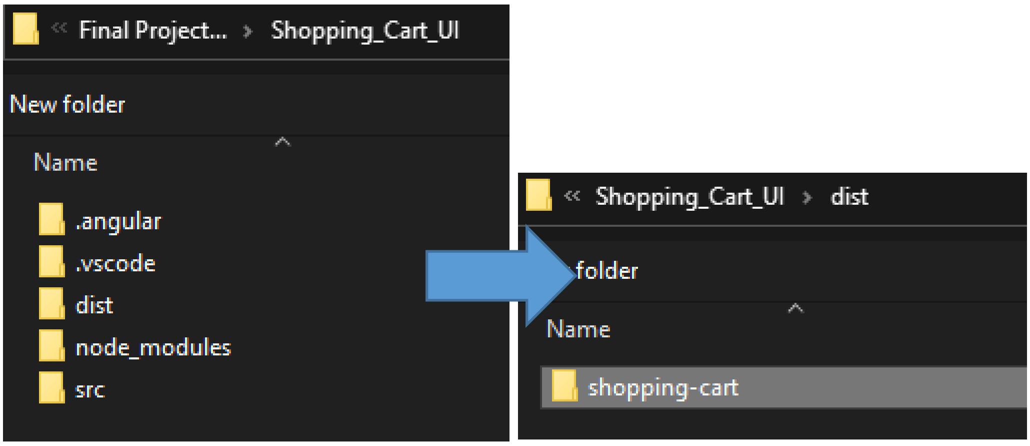
7. Connect with azure account .



8. Select the Web App that we created using runtime stack Node 14, right click and say deploy.

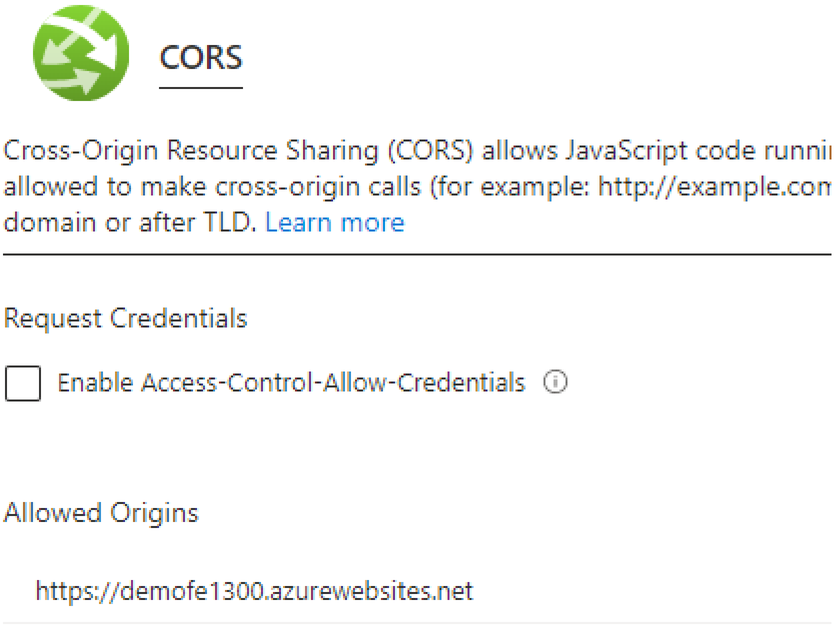


9. When asked which folder to deploy, select folder from dist folder in angular project to deploy .



10. Deploy the project.

11 . Allow URL of Front end in WEB API CORS also.



CORS

Cross-Origin Resource Sharing (CORS) allows JavaScript code runni
allowed to make cross-origin calls (for example: http://example.con
domain or after TLD. Learn more

Request Credentials

☐ Enable Access-Control-Allow-Credentials ⓘ

Allowed Origins

https://demofe1300.azurewebsites.net

12. Now you can run your web Application by clicking URL present in app services