

Appendix A: Deployment Guide

Table of Contents

Prerequisite.....	2
Setup Static Website.....	2
Setup Applications.....	3
Setup URL & API Permission.....	4
Update Config.js file.....	5
Update access-tokens.js file	6
Update appsettings.js file.....	7
Update config-service.js file.....	9
Build the Application	10
Update the deployment.....	12

Deployment Guide for the Azure Function App

Prerequisite

Microsoft Azure Account with access to Administer Active Directory

Microsoft Azure Storage account Resource

Visual Studio Code or something similar for updating the codes

Azure Storage Explorer to upload the code

.Net 6 to build the deployment

Download full source codes from <https://github.com/shiraj-ali/project1>

Setup Static Website

- From all Services, select Storage Account -> then create a storage account with all default settings and enable a static website, type the index name index.html and save.

The screenshot shows the Azure portal interface for managing a storage account named 'rbacstorep0c'. The 'Static website' section is open, showing the following configuration:

- Static website status: Enabled
- Index document name: index.html
- Error document path: index.html

The left sidebar shows other management options like Data management, Shared access signature, Encryption, and Microsoft Defender for Cloud.

- Copy the primary endpoint that will be used in a later step.

The screenshot shows the Azure portal interface for managing a storage account named 'rbacstorepoc'. The 'Static website' section is open, showing the following configuration:

- Static website status: Enabled
- An Azure Storage container has been created to host your static website.
- Primary endpoint: <https://rbacstorepoc.z33.web.core.windows.net/>
- Index document name: index.html
- Error document path: index.html

The left sidebar shows other management options like Data management, Shared access signature, Encryption, and Microsoft Defender for Cloud.

Setup Applications

3. From Main Screen, Select Azure Active Directory -> then select App Registrations

The screenshot shows the Microsoft Azure portal's Default Directory Overview page. In the left sidebar, under 'Azure Active Directory', the 'App registrations' option is selected. At the top, there is a navigation bar with 'Add', 'Manage tenants', 'What's new', 'Preview features', and 'Got feedback?' buttons. Below the navigation bar, a message box displays: 'Microsoft Entra has a simpler, integrated experience for managing all your Identity and Access Management needs. [Entra admin center!](#)'.

4. Select Add then type the display name for the app

The screenshot shows the 'Register an application' page. The 'Name' field is filled with 'rbac-poc'. Under 'Supported account types', the 'Accounts in this organizational directory only (Default Directory only - Single tenant)' radio button is selected. Other options include 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)', and 'Personal Microsoft accounts only'.

5. Take note of the app name and registration details

The screenshot shows the 'rbac-poc' app registration details page. The 'Overview' tab is selected in the left sidebar. The 'Essentials' section displays the following details:

Detail	Value
Display name	rbac-poc
Application (client) ID	d38e30d1-4e61-4b3b-baa1-9cd074f43d83
Object ID	87dfa94a-3371-4afa-9622-3e46d18e0f29
Directory (tenant) ID	5531b16a-ce0e-4f27-bf20-51e2756cfedb
Supported account types	Multiple organizations

On the right side, there are links for 'Client credentials', 'Add a certificate or secret', 'Redirect URIs', '1 web, 0 spa, 0 public client', 'Application ID URI', 'Add an Application ID URI', and 'Managed application in local directory rbac-poc'.

6. register two further applications, Function App – API, Function App Service Principal

The screenshot shows the Microsoft Azure portal interface for managing app registrations. The left sidebar lists various Azure Active Directory components like Users, Groups, External Identities, etc. The main area is titled "Default Directory | App registrations" and shows a list of "Owned applications". There are three entries:

Display name	Application (client) ID	Created on	Certificates
Function App Service Principal	f1519b82-deff-4fa7-9605-67cd25...	11/18/2022	Current
rbac-poc	d38e30d1-4e61-4b3b-baa1-9cd0...	11/14/2022	-
Function App - API	a76ae0b1-d3b8-44c0-b53a-21d4b...	11/18/2022	Current

7. Select Function App – API, select Expose an API, and add a scope “Invoke.API”

The screenshot shows the "Expose an API" configuration for the "rbac-poc" application. The left sidebar has a "Manage" section with "API permissions" selected. The main area shows the "Scopes defined by this API" section, which currently contains "api://rbac-poc/Invoke.API". The "Who can consent" dropdown is set to "Admins only" and the "Admin consent display name" is "Invoke API endpoints".

Setup URL & API Permission

8. Select Authentication and add a platform as a single-page application, then type the Redirect URIs. This was copied in step 2 (from the Storage account Static Website). Also, a local host can be added as http://localhost:300 for testing locally.

9. Then select API permission. Once all the permissions are added, then admin consent is granted.

API / Permissions name	Type	Description	Admin consent req...	Status
user_impersonation	Delegated	Access Azure Service Management as organization users	No	Grant
Invoke.API	Delegated	Invoke API endpoints	Yes	Grant
User.Read.All	Delegated	Read all users' full profiles	Yes	Grant

Update Config.js file

10. Select the overview page, take note of the application (client ID), Directory (tenant) ID and open the source code "project1/client/src/components/config/config.js" and update the client id and tenant id. Also, a redirect URI needs to be added. This will come from step 8 (localhost and storage account static website).

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and a navigation bar with icons for refresh, search, and help. Below the header, the URL is 'All services > Default Directory | App registrations >'. The main title is 'React App'. On the left, there's a sidebar with 'Overview', 'Quickstart', 'Integration assistant', 'Manage', 'Branding & properties', and 'Authentication'. The 'Overview' tab is selected. In the center, under 'Essentials', the app details are listed: Display name: React App, Application (client) ID: 1e09047b-a7b4-4ea9-8d61-af954556e1df, Object ID: e6a1e218-f6fa-4ad4-869d-7e6584db202f, Directory (tenant) ID: 5531b16a-ce0e-4f27-bf20-51e2756cfedb, Client credentials: Add a certificate or..., Redirect URIs: 0 web, 2 spa, 0 public, Application ID URI: Add an Application!, Managed application in I...: React App, and Supported account types: My organization only. Below this, the 'config.js' file is displayed in the Azure Dev Tools for VS Code editor. The code defines msalConfig with auth, clientId, authority, redirectUri, and resource properties.

```

config.js — project1
client > src > components > config > config.js > msalConfig > auth > clientId
1  export const msalConfig = {
2    auth: {
3      clientId: "1e09047b-a7b4-4ea9-8d61-af954556e1df",
4      // This is a URL (e.g. https://login.microsoftonline.com/{your client ID})
5      authority: "https://login.microsoftonline.com/5531b16a-ce0e-4f27-bf20-51e2756cfedb",
6      // This is a URL (e.g. https://login.microsoftonline.com/{your tenant ID})
7      redirectUri:
8        window.location.hostname === "localhost"
9          ? "http://localhost:3000"
10         : "https://rbacstorepoc.z33.web.core.windows.net/",
11      resource: "https://management.azure.com/",
12    },
13    cache: {

```

Update access-tokens.js file

11. Update the access token from the backend application registration, which is Function App – API, to the source code “project1/client/src/utils/access-tokens.js.”

Function App - API | Expose an API

Application ID URI: api://rbac-poc

Scopes defined by this API

```

JS config.js ● JS access-tokens.js ×
client > src > utils > JS access-tokens.js > [o] getAccessTokenScope
1  export const getAccessTokenScope = (type) => {
2    let scope;
3
4    switch (type) {
5      case 'graphApiToken':
6        scope = 'https://graph.microsoft.com/User.Read.All';
7        break;
8
9      case 'managementApiToken':
10        scope = 'https://management.azure.com/user_impersonation';
11        break;
12
13      default:
14        scope = "api://rbac-poc/Invoke.API";
15        break;
16    }
  
```

Update appsettings.json file

12. Update the server-side settings with the tenant name and client ID

Function App - API

Display name: Function App - API

Client credentials: 0 certificate, 1 secret

Redirect URLs: Add a Redirect URI

Application ID URI: api://rbac-poc

Managed application in I... : Function App - API

```

JS config.js ● JS access-tokens.js { appsettings.json ×
server > { appsettings.json > ...
1  {
2    "AzureAd": {
3      "Instance": "https://login.microsoftonline.com/{0}",
4      "Tenant": "shirajenroute.onmicrosoft.com",
5      "ClientId": "f1519b82-deff-4fa7-9605-67cd25178f7f",
6      "ClientSecret": "H1J8Q-iLkXxTou0idWqLSucFwjdtE7HRMyK60cea"
7    },
8    "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;AccountName=rbacstorepoc;
9      AccountKey=N0OPCAKLIS217nzJxeuV2NIhMNoaP6rLjpFIh5V8FWAg5XoJqrVgYmsfuxtsddskmIbl8dTtuvX+AStV2drSA=
10     EndpointSuffix=core.windows.net",
11   }
  
```

13. update the Server side client Secret, which can be set to update every three months

The screenshot shows the Microsoft Azure portal interface for managing a Function App's certificates and secrets. The 'Certificates & secrets' tab is selected. A new client secret has been added with the following details:

Description	Expires	Value	Secret ID
Function App	5/20/2023	MPu*****	f8923605-ec31-447c-805c-e45883024def

The appsettings.json file in the Azure Functions project is open, showing the updated configuration:

```

server > {} appsettings.json ...
1 {
2   "AzureAd": {
3     "Instance": "https://login.microsoftonline.com/{0}",
4     "Tenant": "shirajenroute.onmicrosoft.com",
5     "ClientId": "f1519b82-deff-4fa7-9605-67cd25178f7f",
6     "ClientSecret": "HIJ8Q~iLKxxTou0idWqLSucFwjdtE7HMyK6Qcea"
7   }
  
```

14. update the storage account connection string, which resides in Access keys “AzureWebJobsStorage” and “Configurations.”

The screenshot shows the Microsoft Azure portal interface for managing a storage account named 'rbacstorepoc'. The 'Access keys' tab is selected. The key1 information is displayed:

key1 Rotate key
Last rotated: 18/11/2022 (50 days ago)
Key:
[REDACTED]

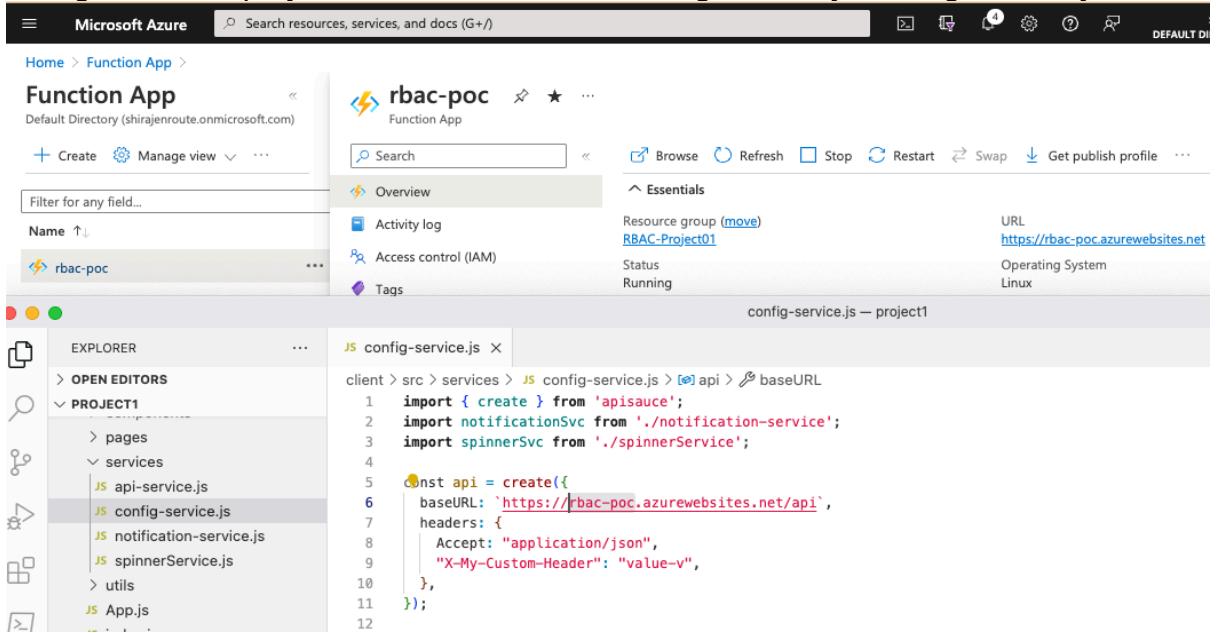
The appsettings.json file in the Azure Functions project is open, showing the updated connection strings:

```

server > {} appsettings.json ...
1 {
2   "AzureAd": {
3     "Instance": "https://login.microsoftonline.com/{0}",
4     "Tenant": "shirajenroute.onmicrosoft.com",
5     "ClientId": "f1519b82-deff-4fa7-9605-67cd25178f7f",
6     "ClientSecret": "HIJ8Q~iLKxxTou0idWqLSucFwjdtE7HMyK6Qcea"
7   },
8   "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;AccountName=rbacstorepoc;
9   AccountKey=N0OPCAKLiS2I7nzJeXeuV2NIHMNoaP6rLjpfIh5V8FWAg5XoJqrVgYmsfuxtsddskmIbl8dTtuvX+AStV2drSA==;
10  EndpointSuffix=core.windows.net",
11   "ConnectionStrings": {
12     "Configurations": "DefaultEndpointsProtocol=https;AccountName=rbacstorepoc;
13     AccountKey=N0OPCAKLiS2I7nzJeXeuV2NIHMNoaP6rLjpfIh5V8FWAg5XoJqrVgYmsfuxtsddskmIbl8dTtuvX+AStV2drSA==;
14     EndpointSuffix=core.windows.net"
15   }
  
```

Update config-service.js file

15. In the front-end, one more detail that needs updating is the URL to call for configuration in “project1/client/src/services/config-service.js” config-service.js



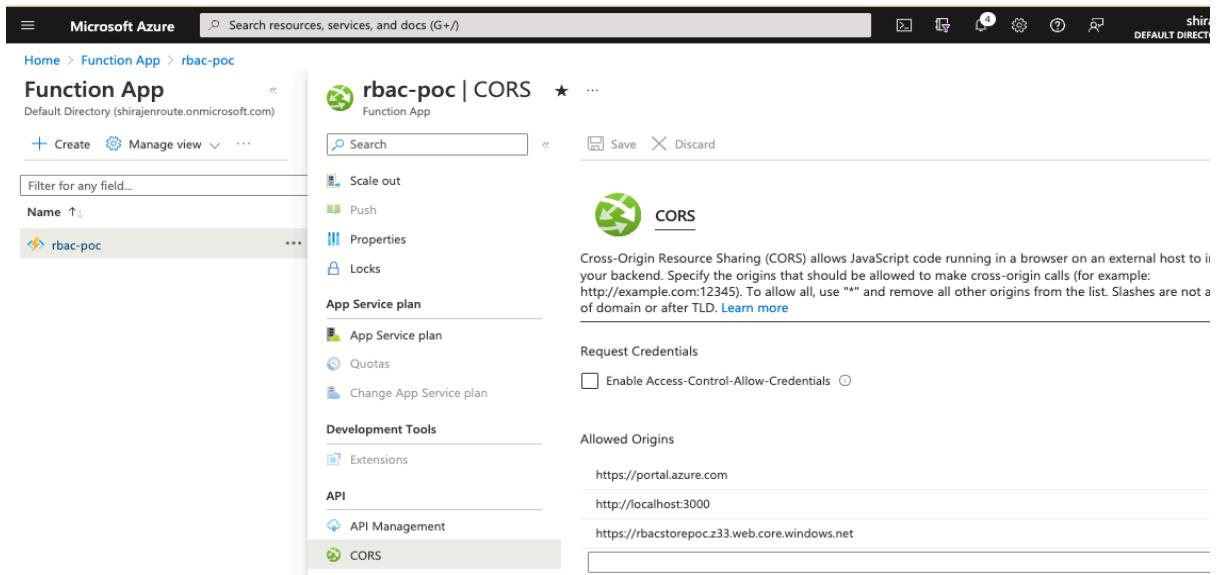
The screenshot shows the Microsoft Azure portal interface for a Function App named "rbac-poc". The left sidebar lists the project structure under "PROJECT1", including files like "api-service.js", "config-service.js", "notification-service.js", "spinnerService.js", and "App.js". The main content area displays the "config-service.js" file content:

```
client > src > services > config-service.js > api > baseURL
1 import { create } from 'apisauce';
2 import notificationSvc from './notification-service';
3 import spinnerSvc from './spinnerService';

4
5 const api = create({
6   baseURL: 'https://[redacted].azurewebsites.net/api',
7   headers: {
8     Accept: "application/json",
9     "X-My-Custom-Header": "value-v",
10   },
11 });

12
```

16. Finally, update the CORS whenever you are calling an API which is not on the same host from the front-end to avoid any errors. Need this enabled for localhost and static web



The screenshot shows the Microsoft Azure portal interface for the "rbac-poc" function app. The left sidebar lists various settings like "Scale out", "Push", "Properties", "Locks", "App Service plan", "Extensions", "API Management", and "CORS". The "CORS" section is currently selected. The right pane displays the CORS configuration settings:

Cross-Origin Resource Sharing (CORS) allows JavaScript code running in a browser on an external host to interact with your backend. Specify the origins that should be allowed to make cross-origin calls (for example: http://example.com:12345). To allow all, use "*" and remove all other origins from the list. Slashes are not a part of domain or after TLD. [Learn more](#)

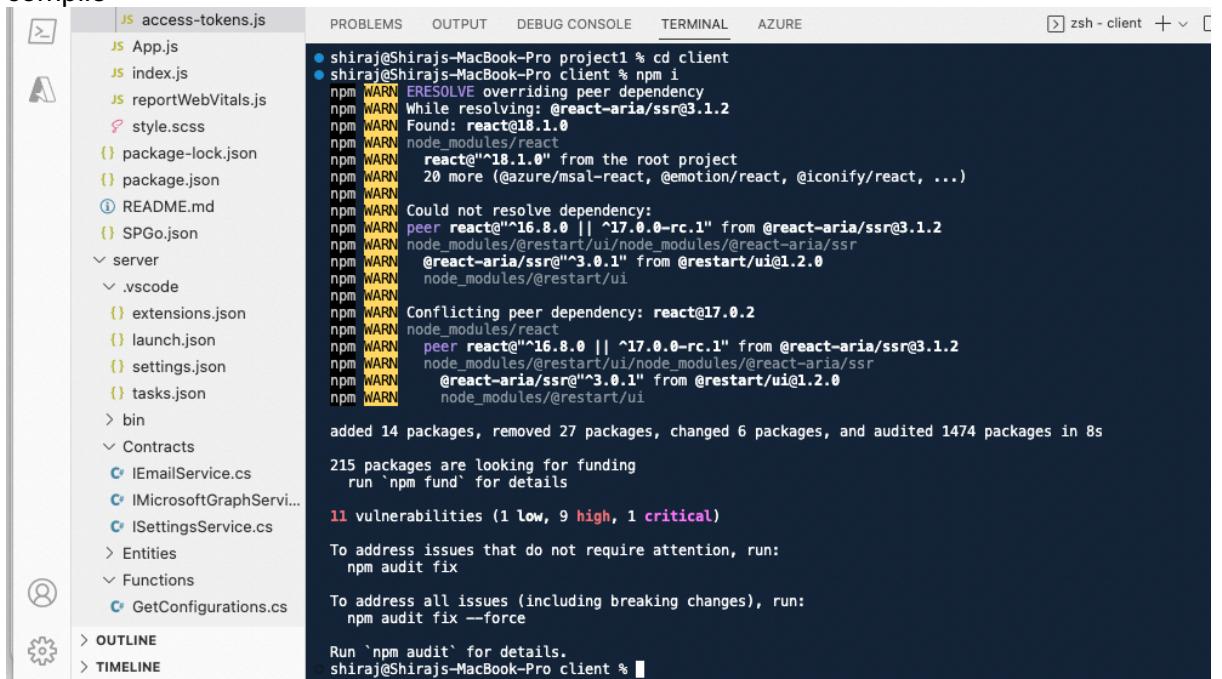
Request Credentials Enable Access-Control-Allow-Credentials

Allowed Origins

- https://portal.azure.com
- http://localhost:3000
- https://rbacstorepoc.z33.web.core.windows.net

Build the Application

17. Now running command “npm i” will download all the node package moudule to compile



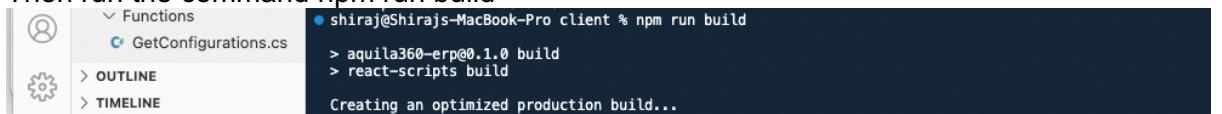
```
shiraj@Shirajs-MacBook-Pro project1 % cd client
shiraj@Shirajs-MacBook-Pro client % npm i
npm WARN ERESOLVE overriding peer dependency
npm WARN While resolving: @react-aria/ssr@3.1.2
npm WARN Found: react@18.1.0
npm WARN node_modules/react
npm WARN   react@"^18.1.0" from the root project
npm WARN   20 more (@azure/msal-react, @emotion/react, @iconify/react, ...)
npm WARN Could not resolve dependency:
npm WARN peer react@"^16.8.0 || ^17.0.0-rc.1" from @react-aria/ssr@3.1.2
npm WARN node_modules/restart/ui/node_modules/@react-aria/ssr
npm WARN   @react-aria/ssr@"^3.0.1" from @restart/ui@1.2.0
npm WARN   node_modules/@restart/ui
npm WARN Conflicting peer dependency: react@17.0.2
npm WARN node_modules/react
npm WARN   peer react@"^16.8.0 || ^17.0.0-rc.1" from @react-aria/ssr@3.1.2
npm WARN   node_modules/restart/ui/node_modules/@react-aria/ssr
npm WARN   @react-aria/ssr@"^3.0.1" from @restart/ui@1.2.0
npm WARN   node_modules/@restart/ui
added 14 packages, removed 27 packages, changed 6 packages, and audited 1474 packages in 8s
215 packages are looking for funding
  run `npm fund` for details
11 vulnerabilities (1 low, 9 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

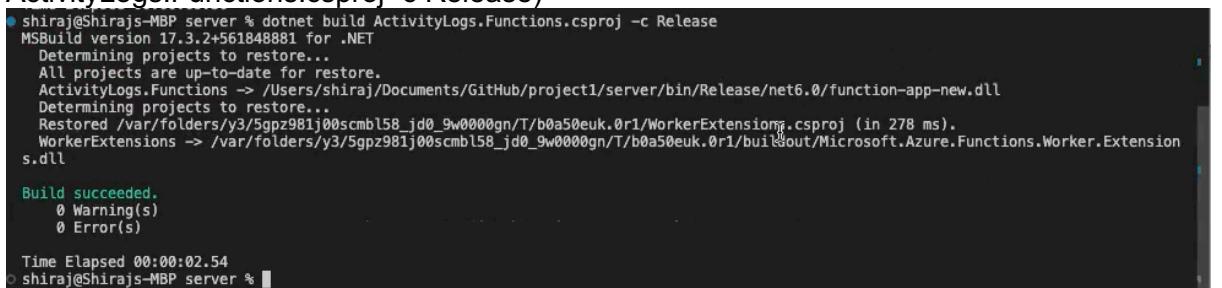
Run `npm audit` for details.
shiraj@Shirajs-MacBook-Pro client %
```

18. Then run the command npm run build



```
shiraj@Shirajs-MacBook-Pro client % npm run build
> aquila360-erp@0.1.0 build
> react-scripts build
Creating an optimized production build...
```

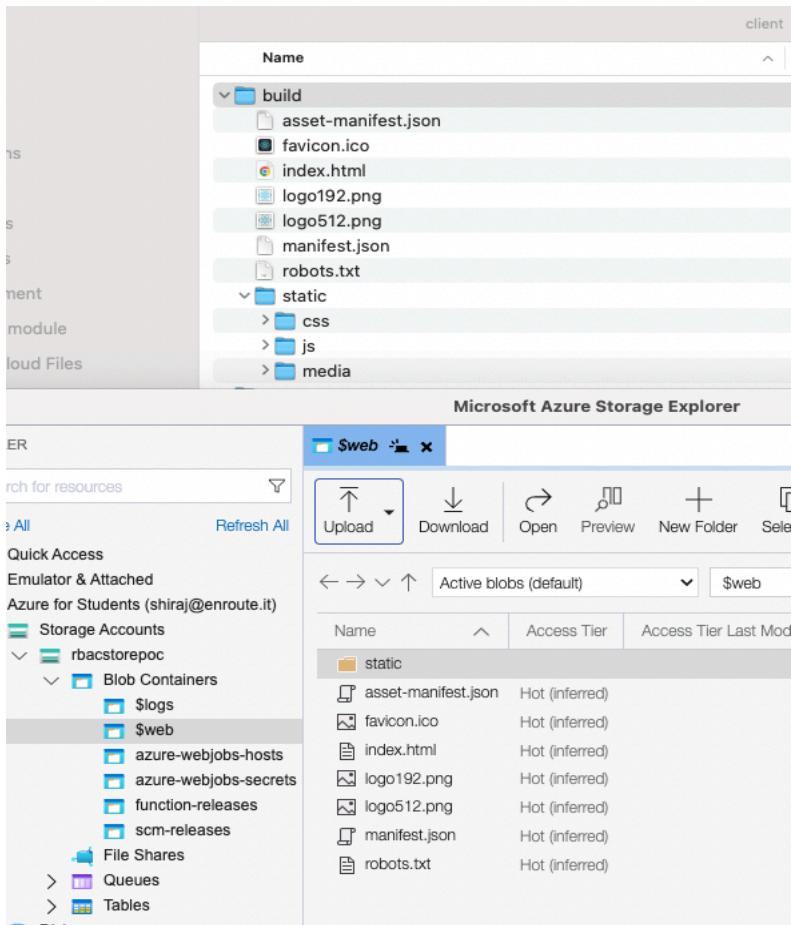
19. Now build for release and deploy the Function App (command to use: dotnet build ActivityLogs.Functions.csproj -c Release)



```
shiraj@Shirajs-MBP server % dotnet build ActivityLogs.Functions.csproj -c Release
MSBuild version 17.3.2+561848881 for .NET
Determining projects to restore...
All projects are up-to-date for restore.
ActivityLogs.Functions -> /Users/shiraj/Documents/GitHub/project1/server/bin/Release/net6.0/function-app-new.dll
Determining projects to restore...
Restored /var/folders/y3/5gpz981j00scmbl58_jd0_9w0000gn/T/b0a50euk.0r1/WorkerExtensions.csproj (in 278 ms).
WorkerExtensions -> /var/folders/y3/5gpz981j00scmbl58_jd0_9w0000gn/T/b0a50euk.0r1/build/Microsoft.Azure.Functions.Worker.Extensions.dll
Build succeeded.
  0 Warning(s)
  0 Error(s)

Time Elapsed 00:00:02.54
shiraj@Shirajs-MBP server %
```

20. Copy all the content from the build folder to web store to make the website live



21. Now need to check the dotnet publish and build to deploy
(command to use: dotnet publish activitylog.functions.csproj -c release -o output/)

```
shiraj@Shirajs-MBP server % dotnet publish ActivityLogs.Functions.csproj -c Release -o output/
MSBuild version 17.3.2+561848881 for .NET
Determining projects to restore...
All projects are up-to-date for restore.
ActivityLogs.Functions > /Users/shiraj/Documents/GitHub/project1/server/bin/Release/net6.0/function-app-new.dll
Determining projects to restore...
Restored /var/folders/y3/5gpz981j00scmbl58_jd0_9w0000gn/T/zacj2tg4.12b/WorkerExtensions.csproj (in 268 ms).
WorkerExtensions > /var/folders/y3/5gpz981j00scmbl58_jd0_9w0000gn/T/zacj2tg4.12b/buildout/Microsoft.Azure.Functions.Worker.Extensions.dll
ActivityLogs.Functions > /Users/shiraj/Documents/GitHub/project1/server/output/
Determining projects to restore...
Restored /var/folders/y3/5gpz981j00scmbl58_jd0_9w0000gn/T/3holsiu3.ugh/WorkerExtensions.csproj (in 140 ms).
WorkerExtensions > /var/folders/y3/5gpz981j00scmbl58_jd0_9w0000gn/T/3holsiu3.ugh/publishout/Microsoft.Azure.Functions.Worker.Extensions.dll
WorkerExtensions > /var/folders/y3/5gpz981j00scmbl58_jd0_9w0000gn/T/3holsiu3.ugh/publishout/
shiraj@Shirajs-MBP server %
```

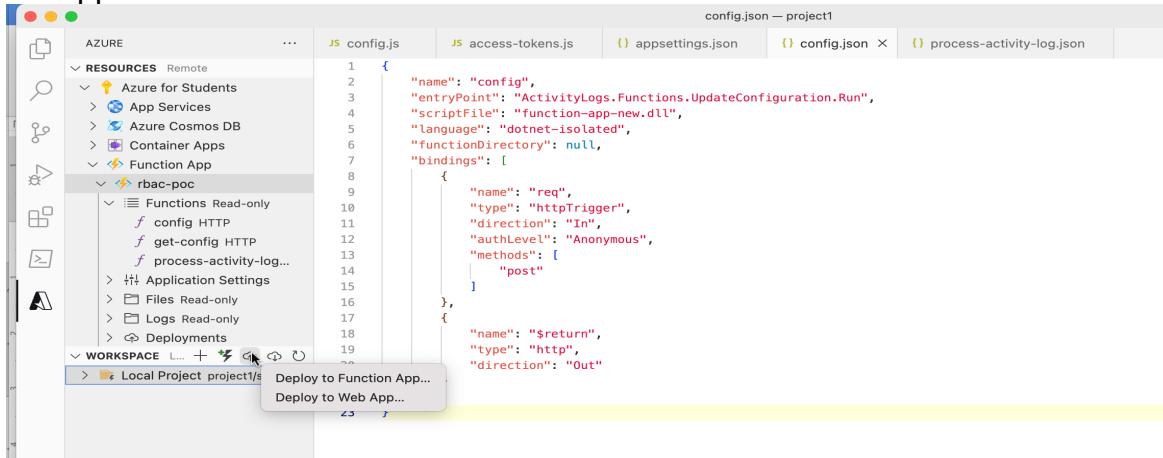
22. Output verification of the deployment

A screenshot of the Azure Functions developer tools interface. The 'OUTPUT' tab is selected, showing deployment logs. The logs show the process of starting deployment, creating a zip package, uploading it to storage, and deploying successfully. It also shows syncing triggers, querying triggers, and listing HTTP trigger URLs. The log entries include:

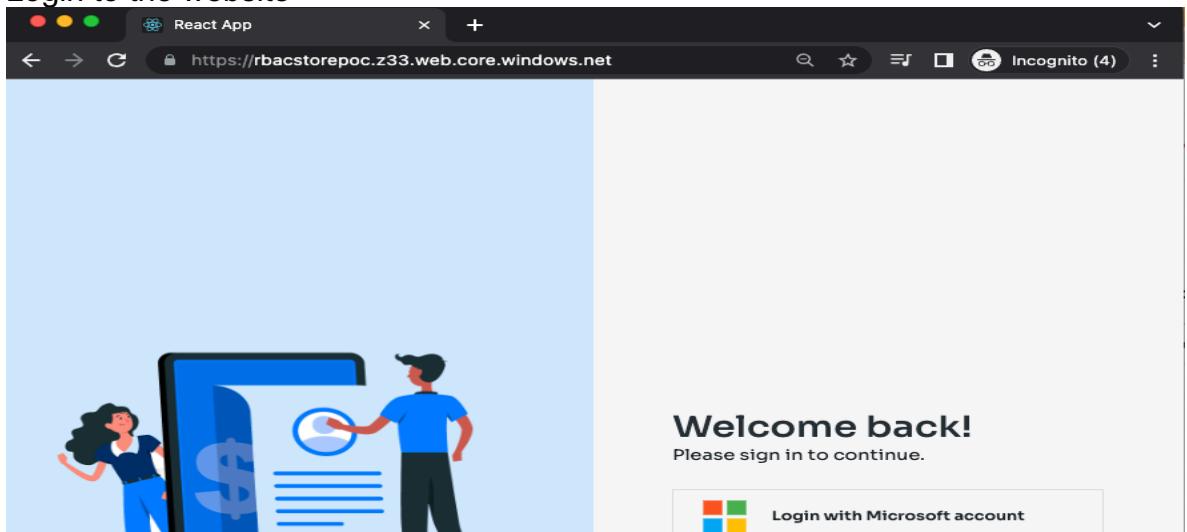
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE
Azure Functions
2:19:09 PM rbac-poc: Starting deployment...
2:19:09 PM rbac-poc: Creating zip package...
2:19:09 PM rbac-poc: Uploading zip package to storage container...
2:19:11 PM rbac-poc: Zip package size: 11.7 MB
2:19:15 PM rbac-poc: Deployment successful.
2:19:25 PM rbac-poc: Syncing triggers...
2:19:28 PM rbac-poc: Querying triggers...
2:19:29 PM rbac-poc: HTTP TriggerUrls:
  config: https://rbac-poc.azurewebsites.net/api/config
  get-config: https://rbac-poc.azurewebsites.net/api/get-config
  process-activity-log: https://rbac-poc.azurewebsites.net/api/process-activity-log
```

Update the deployment

- 23 Connect to the Azure account and update the deployment to the Azure store. This will update everything from the output folder to Azure storage and build the app



- 24 Login to the website



- 25 Finally, verify that all the config settings are getting pulled from the config file once logged in. To use the application, follow Appendix B for RBAC Function User Guide.

