**Providence - Cloud Quest Workshop**

[1. Workshop Objectives: 1](#_Toc2079195450)

[2. Workshop Prerequisites: 1](#_Toc1098434739)

[3. Use Case: 2](#_Toc1194838075)

[4. Step by step implementation guide 2](#_Toc1167438680)

[Section 1: Infrastructure and Code Setup 3](#_Toc1639748775)

[4.1. Steps to connect to Virtual Machine. 3](#_Toc2026293443)

[4.2. Setup GitHub Account 3](#_Toc774930429)

[4.3. Verify Git Installation : 3](#_Toc1575869509)

[4.4. Setup Repos : 4](#_Toc2089772964)

[Section 2: Backend Development 5](#_Toc367033420)

[4.5. Create Function App : 5](#_Toc839456537)

[4.6. Add Connection String in Function App 9](#_Toc562913688)

[Section 3: Frontend Development 11](#_Toc1059266303)

[4.7. User Interface Development 11](#_Toc1423149098)

[4.8. Application Hosting 12](#_Toc282362469)

# **Workshop Objectives:**

* 1. Introduction to Cloud Computing
  2. Develop and deploy Cloud enabled web application to manage health care services.

# **Workshop Prerequisites:**

* + 1. **Installations** : Here are the installations/setup that need to be completed before starting the workshop

|  |  |  |
| --- | --- | --- |
| **S.No** | **Tool** | **Link for installation** |
| 1 | Git Account setup | Setup Git account using the steps mentioned in section [**4.2 : Setup GitHub account**](#_Setup_GitHub_Account)in this document |

* + 1. **Learning Objectives**: Comprehensive understanding of cloud computing concepts, tools, and platforms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Topic Name** | **Topic Components** | **Synchronous/**  **Asynchronous** | **Tutorial Hours** |
| **1.** | Azure Fundamentals | Pre-Read :   * [Azure Fundamentals](https://learn.microsoft.com/en-us/training/paths/microsoft-azure-fundamentals-describe-cloud-concepts/) | Asynchronous | 52 min |
| **2.** | Azure Services | Pre-Read:   * [Azure Functions](https://learn.microsoft.com/en-us/azure/azure-functions/) * [Implement Azure Functions](https://learn.microsoft.com/en-us/training/paths/implement-azure-functions/) * [Azure Data Storage](https://learn.microsoft.com/en-us/training/modules/choose-storage-approach-in-azure/) * [Azure SQL](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql) * [Azure Kubernetes Service (AKS)](https://azure.microsoft.com/en-in/products/kubernetes-service) | Asynchronous | 1 hour |
| **3.** | ReactJs | [Learn React](https://react.dev/) | Asynchronous | 1 hour |

# **Use Case:**

* **Problem Statement:** In today's dynamic health landscape, individuals often face challenges in quickly locating hospitals that not only specialize in specific medical fields like cardiology, orthopedics, or pediatrics but are also within a convenient geographical range.
* **Solution Overview :** There is a need for centralized, easy-to-use platform that aggregates this essential information and personalizes results based on the user's specific needs.

**Healthcare Locator :** Innovative Cloud enabled web application to simplify the process of finding the right hospital based on the specific medical needs and geographic location. Whether you're searching for specialized care in cardiology, orthopedics, pediatrics, or any other field, Healthcare Locator provides a comprehensive and user-friendly platform to connect with the best healthcare options available.

# Step by step implementation guide

# Section 1: Infrastructure and Code Setup

# **Steps to connect to Virtual Machine.**

1. On your local Windows machine, Search for “RDP” in the taskbar.

A screenshot of a computer connection

Description automatically generated

1. In the Remote Desktop Connection window, enter the Virtual Machine name or IP address of your Azure VM in the "Computer" field.
2. Click "Connect".
3. Enter Credentials provided
4. Click "OK".
5. If the credentials are correct, you will be logged into the Windows VM, and the Remote Desktop session will start.

# **Setup GitHub Account**

**NOTE : Skip this step if you already have GitHub account**

1. Visit <https://github.com/signup> and enter all the details  
   A screenshot of a computer

   Description automatically generated
2. Click on continue and complete the verification of your account.
3. After Verification login into Github account created.

# **Verify Git Installation :**

1. Open Git Bash:
   1. Search for "Git Bash" in the Start Menu and open it.
2. Check Git Version:
   1. In the Git Bash window, type command: **git –version**
   2. Press Enter. You should see the Git version number, confirming that Git is installed.

# **Setup Repos :**

* + - * 1. Fork these 2 repos in your account  
           <https://github.com/tech-immersion/HospitalLocator-BE>

<https://github.com/tech-immersion/HospitalLocator-FE>

* + - * 1. Click on Fork Section on top right of the repo  
           A screenshot of a computer

           Description automatically generated
        2. Click on Create Fork  
           A screenshot of a computer

           Description automatically generated
        3. Now you will have a forked version of Both the frontend and backend of the current project.
        4. Open VS code. Click on Terminal option from topbar and choose New Terminal.
        5. Configure git : Execute following commands to configure git in your system
    - git config --global user.name "Your Name"
    - git config --global user.email "your.email@example.com"
      * 1. Clone repository from a remote server (like GitHub) to your local machine using following command .
* git clone forked\_repository\_url
  + - * 1. Navigate to the Repository Directory using following command.
    - cd repository\_name
      * 1. Now you can start developent

# Section 2: Backend Development

# **Create Function App :**

**1. Sign in to Azure Portal**

1. Go to the [Azure Portal](https://portal.azure.com/).
2. Sign in with your given Azure account.
3. **Create a Function App**
4. In the Azure Portal, click on the **"Create a resource"** button again.
5. Search for **"Function App"** and select it.

A screenshot of a computer

Description automatically generated

1. Click on the **"Create"** button.
2. A screenshot of a web page

   Description automatically generatedSelect Consumption option
3. Fill in the required details under the **Basics** tab:
   * **Subscription**: Choose Azure Subscription 1.
   * **Resource group**: Choose **RG-Cloud-Users** resource group
   * **Function App name**: Follow the below naming convention
     + 1. FA-CLOUD-{{UserID}}.
       2. **Ex**- if your userid is cloud-user-1 then your function app name will be **fa-cloud-user1**
   * **Runtime stack**: Select your desired runtime stack (This tutorial uses Node so if you want to continue then Choose **NodeJS**)
   * **Version**: Select the version of the runtime stack. (To follow along with this tutorial, you can choose **20 LTS**)
   * **Region**: Choose **East Asia** in region selection. (Don’t choose any other region deployment might fail)

A screenshot of a computer

Description automatically generated

1. Click on next and go to monitoring tab and disable Application Insights   
   A screenshot of a computer

   Description automatically generated
2. Click **"Review and Create"**.

**4. Review and Create**

1. Review all the settings you have configured.
2. Click the **"Create"** button.
3. Wait for the deployment to complete. You will see a notification once the deployment is successful.

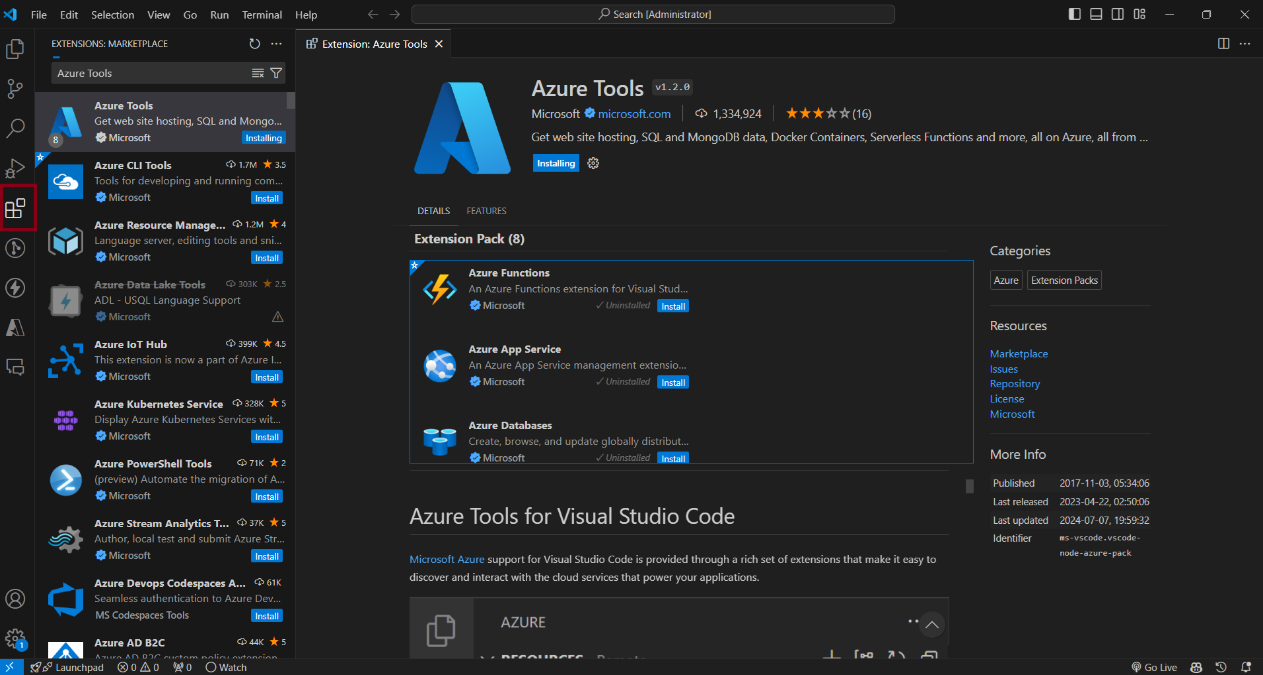
**5. Access Your Function App**

1. Once the deployment is complete, click on **“Go to Resource” .** And You can see your function app.

**6. Deploy Functions App**

* 1. Open **HospitalLocator-BE** code in VS code (Function App code)
  2. Complete “**TO DO”** sections.

File location: getHospitalsList/index.js *line 14*

* + 1. Implement the logic for the sections marked as **TO DO** in the code; to return department details in respective hospitals along with address.
  1. Click on highlighted icon from the side panel. Search **Azure Tools** and click on it. Click on install. Wait for some time till the extension is successfully installed. 
  2. Open Azure tools sidebar in VS Code and click on Sign in to Azure  
     A screen shot of a computer

     Description automatically generated
  3. Complete the Sign in process, make sure you are signing in to the same account where you have created the function app
  4. Now go to the file explorer panel of VS code and right click on **getHospitalsList** function folder and click on Deploy to Function App

A screenshot of a computer

Description automatically generated

* 1. Now select the appropriate subscription and RG and find your deployed function app and select it
  2. Now complete the step and Your code will be successfully deployed to function app in azure
  3. Verify your function (getHospitalList) is visible in bottom section of overview page like the screenshot below, if not redeploy your function app by executing step 6 to 8 again.  
     A screenshot of a computer

     Description automatically generated

# **Add Connection String in Function App**

1. Go to **Function App Overview.**
2. Select **Environment Variables** optionfrom Settings in Left side panel.

A screenshot of a computer

Description automatically generated

1. Click on **Add**. And add Connection String name as **SqlConnectionString** and paste SQL Database connection string for SQL Authentication in Value. Click on **Apply**. **Reach out to Workshop instructor to get the password and replace in the connection strings**

Connection String:

Server=tcp:techimmersionworkshop.database.windows.net,1433;Initial Catalog=techimmersionworkshop;Persist Security Info=False;User ID=techimmersion;Password=<<Ask workshop instructor for password>>**;**MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;

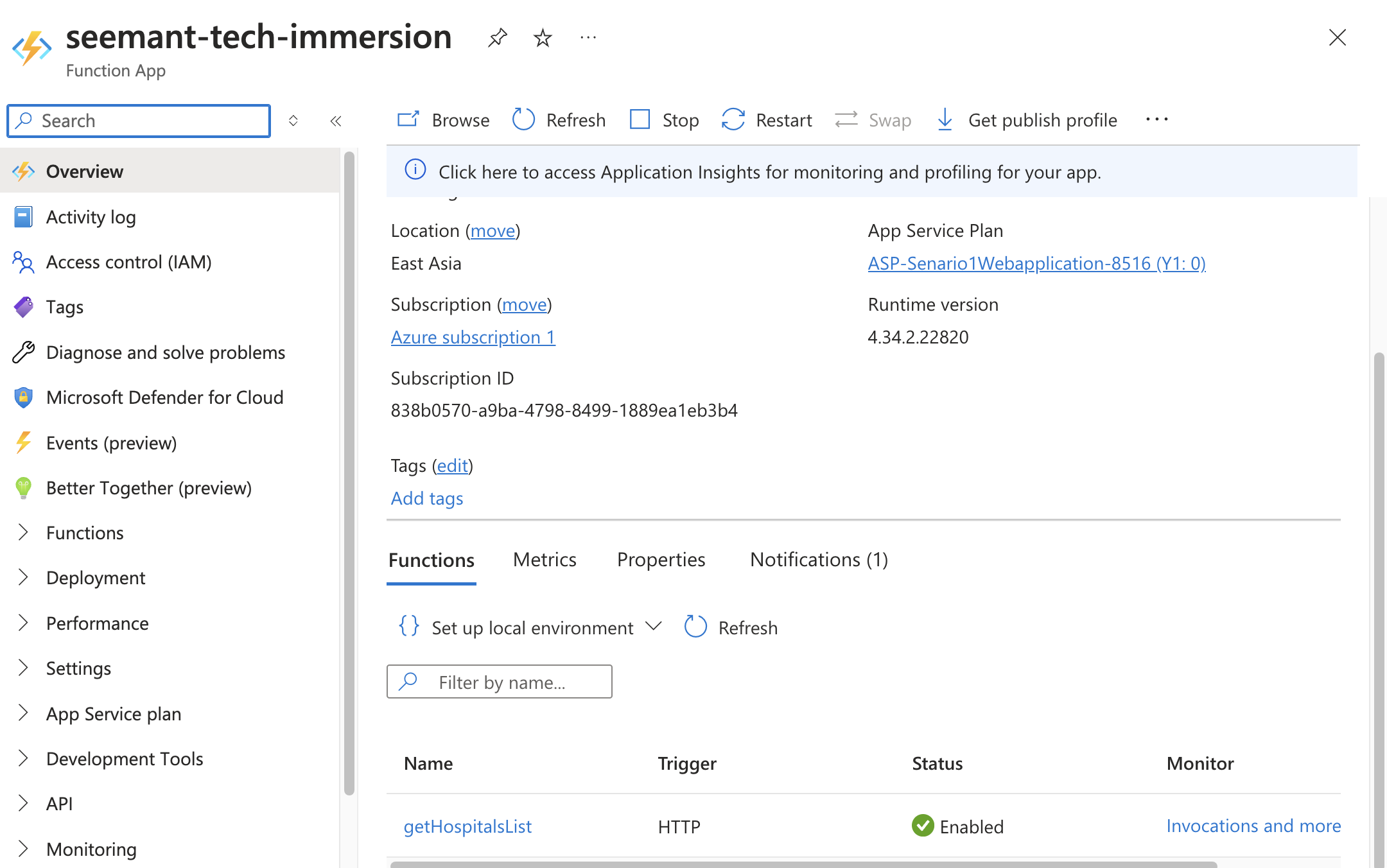
A screenshot of a computer

Description automatically generated

1. Again, Click on **Apply** to restart the function app and to reflect the environment variable updates.
2. Verify whether it is added in App Setting variable list or not.
3. Now in the left panel go to App keys under “Functions” section and create a new host key called anonymous and in value put an empty space and click on add.A screenshot of a computer

   Description automatically generated
4. Now on the left panel search for CORS and click it, and replace the content with “ \* ” like shown in the screenshot below this will enable CORS for all URLS and click on save. A screenshot of a computer

   Description automatically generated
5. Navigate to overview section and click on ***getHospitalsList*** function in bottom section



1. Now to get your API endpoint go to getHospitalsList function and click on get function URL now copy the anonymous URL , this is your **deployed** **REST API endpoint**. You will need this in later steps .  
   A screenshot of a computer

   Description automatically generated

# Section 3: Frontend Development

# **User Interface Development**

1. Open “HospitalLocator-FE “ in VS Code
2. Open a new terminal in VS Code
3. Run this command:

npm install

1. Now all your dependencies will be installed
2. Now run this command

npm start

1. Now your local react server will be up and running and you can visit your localhost URL in browser to see your app running.
2. Now complete the “TO DO” sections and make sure your app is ready to be deployed

There are three TODO section in frontend react app

1. TODO 1: file location : src/index.js line 8

* Go to <https://geocode.maps.co/join/> and create free account
* Now you will get verification email in your email click on the received link to verify yourself.
* Now login using your credentials <https://geocode.maps.co/account/>
* You will see an API key there copy it
* Paste it inside double quotes on line 8 in GEOCODE\_API\_KEY const.

1. TODO 2: file location : src/index.js line 10
   1. Put your **deployed REST API endpoint here**
2. TODO 3: file location src/pages/LocateNearestHospitalMyLoc.js line 356
   1. Make the row.distance as whole number using javascript inbuilt function

# **Application Hosting**

* 1. Go to Azure portal and search for static web apps

A screenshot of a computer

Description automatically generated

* 1. Click on Create button on top left.
  2. Choose subscription as **Azure Subscription 1**
  3. Choose existing resource group [**RG-Cloud-Users**](https://portal.azure.com/#@provtechindia.onmicrosoft.com/resource/subscriptions/838b0570-a9ba-4798-8499-1889ea1eb3b4/resourceGroups/RG-Cloud-Users)
  4. Follow the below naming convention
     + - WA-CLOUD-{{UserID}}.
       - **Ex**- if your userid is cloud-user-1 then your function app name will be **wa-cloud-user1**
  5. Choose plan type as Free
  6. Choose deployment source as GitHub
  7. Click on sign in with GitHub
  8. Authorize your GitHub account
  9. Choose your default organization
  10. Select frontend repo **HospitalLocator-FE**
  11. Select your branch
  12. Now click on next
  13. Choose region as **East Asia**
  14. Click on Review + create and deploy
  15. Now CI – CD will be setup in your repo automatically and on every commit in your selected branch your new code will be deployed
  16. You can find your deployed Front-end URL in static web page overview section like this

A screenshot of a computer

Description automatically generated

* 1. Load the web application using the URL in above screenshot.
  2. Select Department and location to view the nearest hospitals.
  3. Nearest hospital, distance, address and departments should be displayed in the results.
  4. **YOU ARE CLOUD READY!!!!**

THANK YOU for taking part in the workshop. Happy Learning!