

Azure Documentation for Housing Referral Technical Solution

Sept 10, 2024

Provided by:

Alliance Chicago

In support of the Office of the National Coordinator for Health IT Leading Edge Acceleration Projects Grant

Version Table

Version	Date	Changes	Changed By
1.0	July 26, 2024	Rewrite of original EMI Document. Includes all updates to date	Erick Michalski
1.1	Sept 10, 2024	Updates to wording	Erick Michalski

Table of Contents

1	Overview	4
2	Health Data Services and FHIR Resource Setup.....	6
2.1	Setting up Azure Health Data Services Workspace and FHIR Service.....	6
	Deploying Web Applications	8
2.2	Installing Deployment Applications on Local Workstation	8
3	Deploying Service Plan and WebApps in Azure.....	10
3.1	Creating an App Service Plan	10
3.2	Creating Referral (CBO) Application	10
3.3	Creating Backend Service Application	12
3.4	Creating Referral Handler Application	12
4	Configuring Azure Active Directory Authentication for Applications	14
4.1	Registering the Referral (CBO) App in Active Directory.....	14
4.2	Registering the Backend Service.....	16
4.3	Registering the Referral Handler Service	16
4.4	Configuring Referral App Permissions.....	16
5	Configuring the API Management Service.....	17
5.1	Creating the API Management Service.....	17
5.2	Configuring APIs in the Management Service	19
6	Creating the Local Code Repository.....	21
6.1	Referral (CBO) Application.....	21
6.2	Backend Service Application.....	21
6.3	Referral Handler Application.....	21
7	Configuring and Uploading the Application Code	21
7.1	Establish WinSCP connections for the Azure Applications.....	21
7.2	Referral (CBO) Application Deployment	22

7.3	Backend Service Application Deployment.....	23
7.4	Referral Handler Application Deployment.....	25
7.5	Health Data Service	27
8	Testing the Referral Workflow	27
9	Manually Adding Resources to Referral App.....	27
9.1	Adding Practitioners/Providers	27
9.2	Adding Organizations	29
10	Manually Removing Resources to Referral App.....	30
10.1	Adding Practitioners/Providers	30
10.2	Removing Referrals.....	31
11	Updating Applications in Azure.....	31
11.1	Referral (CBO) Application.....	31
11.2	Referral Handler Application.....	32
11.3	Backend Service Application.....	32

1 Overview

This project aimed to address Special Emphasis Notice Area 1: Address health equity and social determinants of health through innovative, open-source technology tools, and electronic health records.

1. Understand and characterize the elements of a comprehensive care plan bridging the domains of need and services across both organizations,
2. Co-design and pilot test standards-based, open source, FHIR-enabled electronic interoperable care plan accessible to both institutions and the patient, and
3. Conduct a rigorous evaluation of the pilot approach to understand its impact on the and considerations for scaling across additional service domains and institutions.

The team designed a referral infrastructure system to support the “Aligning Housing and Healthcare” use cases between a community health center and a community-based organization (CBO), herein referred to as the eCare Plan.

This document details the setup of Azure Health Data Services (AHDS) and FHIR Services, the Azure Active Directory, and deployment of the CBO and Referral Handler Service applications. It also reviews the set up and hosting of the Referral Handler Service at a Fixed IP Address. Finally, it establishes the connection between the Referral Handler Service to Athena Practice (aP) for authentication and reading of FHIR data. It does not include documentation on the authentication and connection of the Referral Handler Service to AHDS and of the eCare Plan to AHDS.

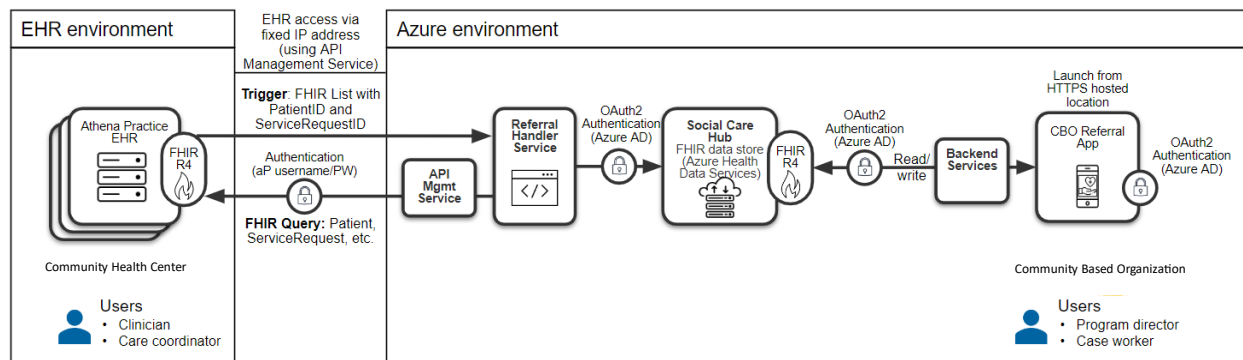
Architecture Diagram

The following is an architecture diagram (Figure 1) of the entire referral infrastructure system. Here is a short description of each component:

- **Athena Practice EHR:** This is the EHR system that Community Health Center (CHC) uses. It has FHIR R4 API endpoints to expose certain FHIR Resources to external, authorized and authenticated entities. CHC users include clinicians and care coordinators.
- **Referral Handler Service:** This is a stateless backend service that is hosted on the Azure secure cloud environment. It is hosted at a fixed IP address using an API management service. The IP has been whitelisted by athenaPractice. It is capable of querying and receiving FHIR payloads and saving it to a FHIR data store.
- **Azure API Management Service:** An Azure service to manage APIs in a cloud environment. Azure API Management Service has a Static IP, which is whitelisted and is used to manage the APIs for requesting access and query the FHIR Resources from Athena Practice EHR.
- **Social Care Hub/ Azure Health Data Services (AHDS):** Azure Health Data Services is a cloud based FHIR data store hosted in the Azure environment. The project has called this data store the “Social Care Hub”. This is where data authored by the eCare Plan is stored.
- **Backend Services:** This is the backend service for the eCare Plan. It performs operations such as querying, creating, saving, and updating FHIR resources in the Social Care Hub.
- **eCare Plan:** The eCare Plan is the user-facing application that allows both the community-based organization and CHC users to see and interact with the Referral information stored in the Social Care Hub.

- Azure Active Directory:** Azure Active Directory (recently renamed “Microsoft Entra ID”) is a cloud-based identity and access management solution. CBO and CHC users who should have access to the eCare Plan are added by the Azure administrator to the Azure Active Directory. The users use an email and password which is validated by Azure AD to get access to the eCare Plan.

Figure 1 AC LEAP Architecture Master Diagram



2 Health Data Services and FHIR Resource Setup

2.1 Setting up Azure Health Data Services Workspace and FHIR Service

The first part of this deployment is the configuration of the environment which contains the Azure Health Data Services (AHDS) workspace. Within this workspace a custom template was used to enable the Azure Health Data Service workspace, FHIR Service, FHIR-Proxy function app, FHIR Loader function app, App Service Plan, Blob Storage for FHIR service, Storage Account for FHIR Proxy function app, Storage Account for FHIR Loader, Key Vault, Log Analytics Workspace, App insights for FHIR Loader app, App Insights for FHIR Proxy app, Event Grid System, and a Redis Cache. The for the custom Azure template is provided below.

- <https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2Fmicrosoft%2FAzure-health-data-services-workshop%2Fmain%2Fresources%2Fdeploy%2Fdeployfhirtrain.json>

Only the Resource Group, Region and Deployment prefix would normally be verified/changed. A specific prefix of “acleap” was used for the AllianceChicago deployment.

Figure 2: Custom Deployment in Azure

Home >

Custom deployment

Deploy from a custom template

New! Deployment Stacks let you manage the lifecycle of your deployments. Try it now →

Resource group * ⓘ [Create new](#)

Instance details

Region * ⓘ	<input type="text" value="(US) East US"/> ✓
Resource Tags ⓘ	<input type="text" value='{"environmentName":"Azure Health Data Services OpenHack","challen...'/> ✓
Deployment Prefix * ⓘ	<input type="text" value="emi"/> ✓
Fhir Server Tenant Name ⓘ	<input type="text" value="[subscription().tenantId]"/>
Resource Location ⓘ	<input type="text" value="[resourceGroup().location]"/>
Enable Consent Opt Out ⓘ	<input type="text" value="true"/>
Enable Date Sort ⓘ	<input type="text" value="false"/>
Enable Participant Filter ⓘ	<input type="text" value="false"/>
Enable Fhir Cds Sync Agent ⓘ	<input type="text" value="false"/>
Enable Publish Fhir Event ⓘ	<input type="text" value="false"/>
Enable Profile Validation ⓘ	<input type="text" value="false"/>
Enable Transform Bundle ⓘ	<input type="text" value="true"/>
Enable Everything Patient ⓘ	<input type="text" value="false"/>
Use MSI ⓘ	<input type="text" value="true"/>
App Service Plan Sku	<input type="text" value='{"name":"EP1","tier":"ElasticPremium","size":"EP1","family":"EP"}'/> ✓

(The deployment of the workspace using the provided template did generate some errors/failures related to the smart detector rules and the pxfa function app. As those components are not used, the errors can be ignored.)

After completing this template, the resource group is deployed with the necessary components to start storing data in the storage containers which can then be accessed through the FHIR Service. As noted above, not all of the default AHDS components are utilized by this deployment.

Table 1: Components for Resource Group

COMPONENT	DESCRIPTION
Azure Health Data Services workspace	This is a managed Platform as a Service (PaaS) which provides a dedicated environment for health-related data management and operations.
FHIR service	A managed Fast Healthcare Interoperability Resources (FHIR) server which standardizes the exchange of healthcare information.
FHIR-Proxy (OSS) Function App	An Open-Source Software (OSS) function which filters FHIR data input/output.
FHIR Loader (OSS) Function App	This function is responsible for ingesting or importing FHIR data into the system.
App Service Plan	A shared environment which hosts both the FHIR-Proxy and FHIR Loader function apps, providing them with the necessary resources to run.
Storage account (Blob storage for FHIR service)	A storage account needed to hold blob data for the FHIR service.
Storage account (for FHIR-Proxy function app)	Dedicated storage for the FHIR-Proxy function app to save and manage its data.
Storage account (for FHIR Loader)	Dedicated storage for the FHIR Loader function to save and manage its data.
Key Vault	A secure vault which stores secrets and configuration settings. This ensures that sensitive data like API keys or database connection strings are kept safe and away from the main application code.
Log Analytics Workspace	A monitoring tool which captures and logs the activity of the deployed components, providing insights and analytics.
Application Insights (for FHIR Loader application)	Monitoring service specifically for the FHIR Loader application, allowing for performance tracking, logging, and troubleshooting.
Application Insights (for FHIR-Proxy application)	Like the above but tailored for the FHIR-Proxy application.
Event Grid System Topic	A messaging component that triggers the processing of FHIR bundles placed in the FHIR Loader storage account.
Redis Cache	An in-memory data structure store, required by the FHIR-Proxy modules, specifically for operations like Consent Opt Out.

Deploying Web Applications

2.2 Installing Deployment Applications on Local Workstation

In order to deploy code in the Azure hosted environment, the following applications will need to be deployed on the local system

- WinSCP
- Git
- NPM
- Yarn
- VScode(or any other IDE) in your system
- Postman

WinSCP

- Download WinScp from https://winscp.net/eng/download.php#google_vignette
- Install application using the downloaded install application

GIT

- Download Git:
 - Go to the official Git website: <https://git-scm.com/>
 - Look for the download section on the homepage.
 - Click on the appropriate download link for your operating system. Git is available for Windows, macOS, and Linux.
- Install Git:
 - For Windows:
 - Once the download is complete, locate the downloaded .exe file and double-click on it to start the installation process.
 - Follow the installation wizard instructions. You can usually accept the default settings unless you have specific preferences.
 - During the installation, you might be asked to select components to install. Keep the default selections unless you have a specific reason not to.
 - After the installation is complete, Git should be ready to use from the command line. You can open a command prompt or Git Bash to verify the installation by typing "git – version".

NVM

- To download and install NVM on Windows, follow these steps:
 - Download: Go to <https://github.com/coreybutler/nvm-windows/releases>
 - Download nvm-setup.exe
 - Run nvm-setup.exe
 - Reboot to set the variables and path correctly

NPM

- To download and install NPM (Node Package Manager), you typically install Node.js, which includes NPM as part of its installation. Below are the steps to download and install Node.js with NPM on Windows:
- Download Node.js Installer:

- Visit the official Node.js website: <https://nodejs.org/en/download/package-manager>
 - Download the Windows installer (.msi file), either the LTS (Long-Term Support) version or the current version. (v16.20.2 is the one actually used)
- Run the Installer:
 - Double-click the downloaded installer file to run it.
 - Follow the installation wizard instructions.
 - Leave the default options selected unless you have specific requirements.
- Verify Installation:
 - Open a command prompt or PowerShell.
 - Type `node -v` and press Enter to check if Node.js is installed.
 - Type `npm -v` and press Enter to check if NPM is installed.

VSCode

- To download and install Visual Studio Code (VSCode) on Windows, follow these steps:
 - Download: Go to <https://code.visualstudio.com/>, click "Download for Windows".
 - Install: Open the downloaded file and follow the installation instructions.
 - Launch: Once installed, open Visual Studio Code from the Start menu or desktop shortcut.

Yarn

- From command prompt, run
 - `Npm install -global yarn`
- From command prompt
 - `Yarn -version` (should be 1.22.22)
 - `Yarn install`

Postman

- <https://www.postman.com/downloads/>

3 Deploying Service Plan and WebApps in Azure

3.1 Creating an App Service Plan

Navigate to the Azure tenant where the Azure Health Data Services workplace was deployed. Select "+ Create a resource", search for "App Service Plan", and select "Create". Fill in details like subscription, resource group name (from Step 2), service plan name, operating system choice (Windows), region (preferably the same as your resource group-North Central), and the pricing tier. Once reviewed, click "Create".

Create App Service Plan ...

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Azure Corp Subscription

Resource Group * ⓘ AC_LEAP
[Create new](#)

App Service Plan details

Name * ACLeap-appserviceplan ✓

Operating System * ☐ Linux ☒ Windows

Region * North Central US

Pricing Tier

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#) ⓘ

Pricing plan Standard S1 (100 total ACU, 1.75 GB memory, 1 vCPU)
[Explore pricing plans](#)

Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed. [Learn more](#) ⓘ

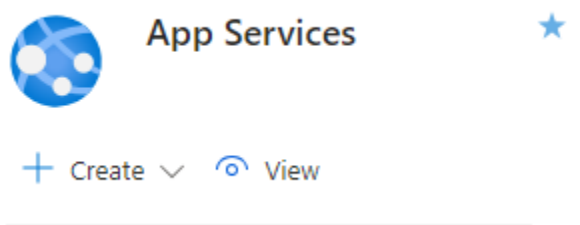
Zone redundancy ☐ Enabled: Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three. ☒ Disabled: Your App Service Plan and the apps in it will not be zone

3.2 Creating Referral (CBO) Application

Step 1:

- Login into Azure portal and search for "App Services" Resource at the top.

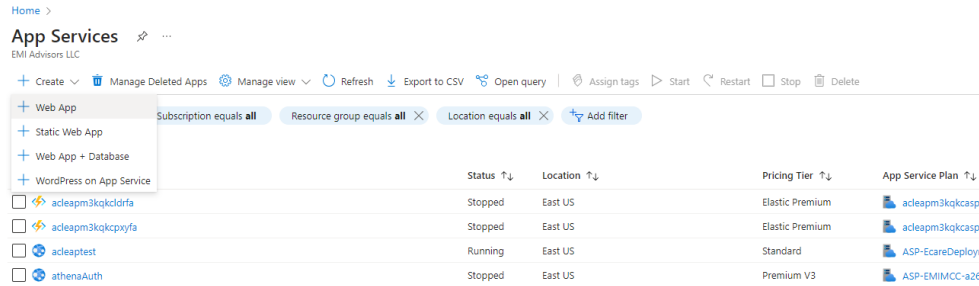
Image 1: Azure Portal



Step 2:

- In App Services page, click on “Create” option (located on top left side of the window) and select "Web App" from the drop-down menu.

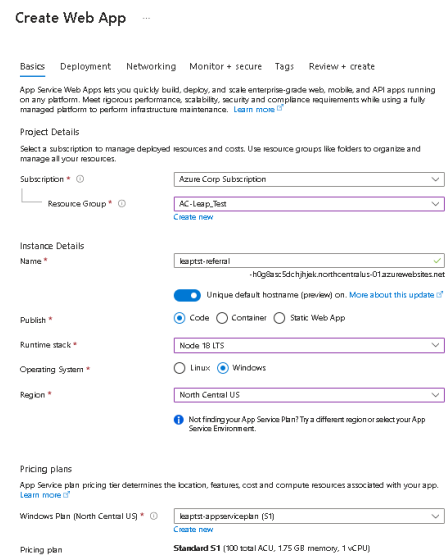
Image 2: App Services



Step 3:

- Select your Subscription and your Resource group from the drop-down menu.
- Enter the name of the application (acleap-referral).
- Select “Code” for publish.
- For Runtime stack, select “Node 18 LTS” from the drop-down menu.
- **Select Operating System as “Windows”.**
- Select the region, based on respective location.
- The remaining default settings should be sufficient unless something custom is needed.
- “Review + create”.

Image 3: Creation of a Web App



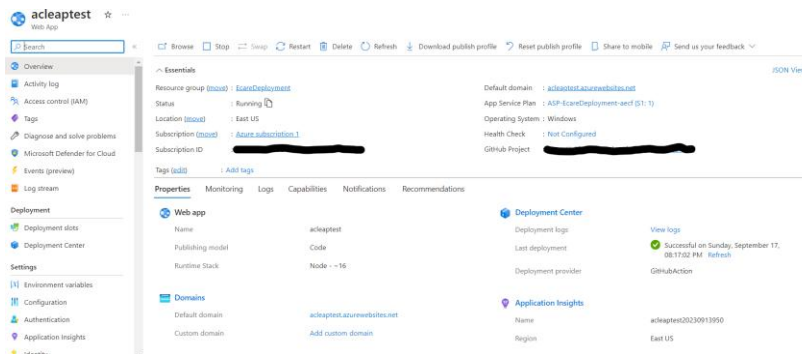
Step 4:

- If all the parameters are correct, select “Create”, that creates the app services.

Step 5:

- After the app service creation process completes, the overview page will be displayed.

Image 4: App Services overview page



3.3 Creating Backend Service Application

Step 1:

- Login into Azure portal and search for “App Services” Resource.

Step 2:

- In App Services page, click on “Create” option (located on top left side of the window) and select "Web App" from the drop-down menu.

Step 3:

- Select your Subscription and your Resource group from the drop-down menu respectively.
- Enter the name of the application. (acleap-backend)
- Select “Code” for publish.
- For Runtime stack, select “Node 18 LTS” from the drop-down menu.
- **Select Operating system as “Linux”.**
- Select the region, based on respective location.
- The remaining default settings should be sufficient unless something custom is needed.
- “Review + create”.

Step 4:

- If all the parameters are correct, select “Create”, that creates the app services.

Step 5:

- After the app service creation process completes, the overview page will be displayed.

3.4 Creating Referral Handler Application

Step 1:

- Login into Azure portal and search for “App Services” Resource.

Step 2:

- In App Services page, click on “Create” option (located on top left side of the window) and select "Web App" from the drop-down menu.

Step 3:

- Select your Subscription and your Resource group from the drop-down menu respectively.
- Enter the name of the application. (acleap-referral-handler)
- Select "Code" for publish.
- For Runtime stack, select "Node 18 LTS" from the drop-down menu.
- Select Operating system as "Linux".
- Select the region, based on respective location.
- The remaining default settings should be sufficient unless something custom is needed.
- "Review + create".

Step 4:

- If all the parameters are correct, select "Create", that creates the app services.

Step 5:

- After the app service creation process completes, the overview page will be displayed.

4 Configuring Azure Active Directory Authentication for Applications

Reference: Creating Azure AD registrations for the applications and integrating the registrations into the code of the applications. **Each of the three web applications created in section 3 will need to be registered in Azure.**

4.1 Registering the Referral (CBO) App in Active Directory

Step 1:

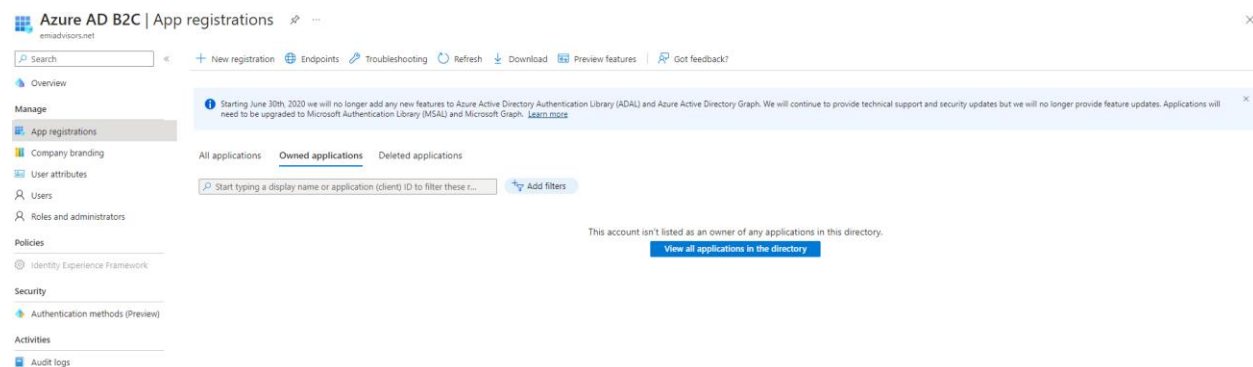
- Search for the Azure AD B2C resource in Azure and select it



Step 2:

- Under Manage, locate "App registrations" on the left-hand side of the menu.

Image 5: App registrations



Step 3:

- Click on "+ New registration"
- Enter the name of the application.
 - acleap-referral
- Under "Supported account types", select "Accounts in this organizational directory only ([name of organization] only - Single tenant)"
- Under Redirect URI, select "Web" in the dropdown and enter the redirect URL as "https://<yourappservice>.azurewebsites.net/" (This URL should correspond to the deployed URL of that respective application and can be found under the Default Domain heading on the overview page of that application.).
- After filling in all the required details, click on "Register" to register the application in Azure AD.

Image 6: Register an Application

All services > Azure AD B2C | App registrations >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

Supported account types

Who can use this application or access this API?

☒ Accounts in this organizational directory only (EMI Advisors LLC only - Single tenant)

☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)

☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

☐ Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Select a platform

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

[Register](#)

Step 4:

- Once the app registration is created in Azure AD, navigate to the overview page where you will find ClientID, TenantID, and Redirect URLs.

Image 7: ClientID, TenantID, and Redirect URLs

Home > Azure AD B2C | App registrations >

ACLeapDevLocal

Search

Overview Quickstart Integration assistant Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions Expose an API App roles Owners Roles and administrators Manifest Support > Troubleshooting Troubleshooting New support request

Delete Endpoints Preview features

Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Essentials

Display name	: ACLeapDevLocal	Client credentials	: Add a certificate or secret
Application (client) ID	: [REDACTED]	Redirect URIs	: 1 web, 0 spa, 0 public client
Object ID	: [REDACTED]	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: [REDACTED]	Managed application in L...	: ACLeapDevLocal
Supported account types	: My organization only		

Starting June 30th, 2022 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

[Get Started](#) Documentation

Build your application with the Microsoft identity platform

The Microsoft identity platform is an authentication service, open-source libraries, and application management tools. You can create modern, standards-based authentication solutions, access and protect APIs, and add sign-in for your users and customers. [Learn more](#)

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

[View API permissions](#)

Sign in users in 5 minutes

Use our SDKs to sign in users and call APIs in a few steps. Use the quickstarts to start a web app, mobile app, SPA, or daemon app.

[View all quickstart guides](#)

Configure for your organization

Assign users and groups, apply conditional access policies, configure single sign-on, and more in Enterprise applications.

[Go to Enterprise applications](#)

Step 5:

- Copy the ClientID, TenantID, and Redirect URL from the Azure application.
 - These will be needed later for modifying the code in each application deployment

Step 6:

- In the left pane, choose authentication
- Under Implicit grant and hybrid flows, select Access tokens and ID tokens
- Save

4.2 Registering the Backend Service

- Follow the registration steps in section 4.1 to register the Backend Service application
- Once the app is registered, go into the app registration
- In the left tab, select Certificates & Secrets
 - New client secret
 - Add description
 - Click on Add
- Note the Value as it will be used for Backend Service .env file configuration

4.3 Registering the Referral Handler Service

- Follow the registration steps in section 4.1 to register the Referral Handler application
- Once the app is registered, go into the app registration
- In the left tab, select Certificates & Secrets
 - New client secret
 - Add description
 - Click on Add
- Note the Value as it will be used for Backend Service .env file configuration

4.4 Configuring Referral App Permissions

- In Azure portal, search for Microsoft Entra ID and open it
- In the left pane, select Manage and then Groups
- Create a new Security group to be used for permissions
- In the left pane, under Manage, select Enterprise Applications
- Select the referral application
- Under Assign Users and Groups, add the permissions security group created above
 - For the role, leave it as Default Access
- In the left tab choose properties and change the assignment required parameter to yes
- Save the configuration
- Add users to the group to facilitate access to the referral application

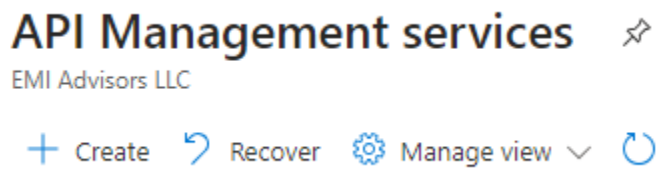
5 Configuring the API Management Service

5.1 Creating the API Management Service

Step 1:

- In the Azure tenant, search for “API Management service”
- Select “Create”.

Image 8: API Management Services



Step 2:

- Enter all the required fields (Subscription, Resource Group, Region, Resource Name, Organization Name and Administrator email with their respective)
 - ***It is important that you use the DEV pricing plan or the standard (not v2) pricing plan. The default standard plan setting may not generate a static IP address.***
- Select “Review + Create” button.

Image 9: Create API Management Service

Create API Management service ...

API Management service

Basics Monitoring Scale Managed identity Virtual network Protocol settings Tags Review + install

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Resource group * ⓘ [Create new](#)

Instance details

Region * ⓘ

Resource name *

Organization name * ⓘ

Administrator email * ⓘ

Pricing tier

API Management pricing tiers vary in computing capacity per unit and the offered feature set - for example, support for virtual networks, multi-regional deployments, or self-hosted gateways. To accommodate more API requests, consider adding API Management service units instead. [Learn more](#)

⚠ The Developer tier of API Management does not include SLA and should not be used for production purposes. Your service may experience intermittent outages, for example during upgrades. [Learn more](#)

Pricing tier ⓘ [See all pricing tiers](#)

Step 3:

- Verify values and select “Create”.

Image 10: Create API Management Service

Home > API Management services >

Create API Management service ...

API Management service

Basics

Subscription	[REDACTED]
Resource group	[REDACTED]
Region	[REDACTED]
Resource name	[REDACTED]
Organization name	[REDACTED]
Administrator email	[REDACTED]
Pricing tier	Developer

Monitoring

Application Insights	Disabled
Application Insights instance	-

Scale

Unit(s)	1
---------	---

Managed identity

Identity type	None
---------------	------

Network

Connectivity type	None
-------------------	------

Protocol settings

Triple DES (3DES)	Disabled
HTTP/2	Disabled
TLS 1.1 (HTTP/1.x only)	Disabled
TLS 1.0 (HTTP/1.x only)	Disabled
SSL 3.0 (HTTP/1.x only)	Disabled
TLS 1.1	Disabled

[Create](#) [< Previous](#) [Next >](#) [Download a template for automation](#)

Step 4:

- After creating the API Management Services, open the overview page
- The “Virtual IP (VIP) address” is a fixed IP address,
 - This is the IP that aP will use to generate referrals

Image 11: Virtual IP Addresses

acleanpreferrallhandler API Management service

Search

Developer portal Delete Open in mobile

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Platform migration

Essentials

Resource group (moad)

Status : Online

Location : East US

Subscription (moad)

Subscription ID

Tags (edit) : Add tags

Developer portal URL

Gateway URL

Tier : Standard

Virtual IP (VIP) addresses

Platform version : stv2

Properties Get started Learn more Monitor Recommendations (0)

Princine tier

Delegation

JSON View

5.2 Configuring APIs in the Management Service

Step 1

- From the left panel of the API Management Service Screen, select Subscriptions under APIs
- Select “+ Add subscription”
 - Name: ACLearnReferral
 - Display name: [same as above]
 - Everything else can be left blank
 - Create
- Show/Hide keys
- Note the Primary Key – It will be copied into referral handler .env file

Step 2

- From the left panel on the API Management Service Screen, select APIs
- Using the “+ Add API” option and add 2 APIs to the configuration
 - HTTP
 - AthenaAppXX (using XX health center identifier)
 - Create
 - Settings
 - Web Service URL –
<https://apXX.domain.org:9443/XXAPIServer/oauth2>
 - URL Suffix - connectXX
 - Create
 - Design – add operation
 - Frontend
 - AthenaAppXX-get-token
 - URL – Get /token
 - Save
 - Inbound Processing
 - “+ Add policy” > Set Query Parameters
 - Name – scope
 - Value – user%2F*.*
 - Action – append
 - Name – grant_type
 - Value – client_credentials
 - Action – append
 - Save
 - “+ Add policy” > Set Header
 - Name – authorization
 - Value – Basic aGguYXBpXXXXXXXXXX
 - This is the Base64-encoded value for the user ID & password used in authenticating to the athenaPractice
 - fhir server interface. It can be generated using Notepad ++ by:
 - openblank page

- enter "Username":"password".
 - Select (highlight)
"Username":"password"
 - From the top ribbon, choose
Plugins/Mime Tools/Base64
Encode
 - Copy hash value to the Policy
value above (preceded by
Basic).
 - Action – append
 - Save
- Settings tab on the top
 - Base URL - copy to athena fhir url for the referral handler .env
file
- AthenaAppResources
 - Create
 - Settings
 - Web Service URL – <https://apXX.domain.org:9443/XXAPIServer/fhir/r4>
 - URL Suffix - hresources
 - Save
 - Design – add operation
 - Fontend
 - ResourcesAthena
 - URL – Get /ServiceRequest
 - Save
 - Inbound Processing
 - Set Query Parameters
 - Name – _count
 - Value – 10000
 - Action – append
 - Name – _format
 - Value – json
 - Action – append
 - Name – _summary
 - Value – data
 - Action – append
 - Name – category
 - Value – referrals
 - Action – append
 - Name – _include
 - Value – ServiceRequest:*
 - Action – append
 - Save
 - Settings tab

- Base URL - copy to athena fhir url for the referral handler .env file
- Go to Overview
 - Copy gateway URL – will go into referral handler service .env file

6 Creating the Local Code Repository

Reference: Initial code for the three application services will be copied from a central GitHub repository.

6.1 Referral (CBO) Application

- On a local device, create a directory to hold the repository
- Change path to the target local directory
- Login to GitHub
- Navigate to <https://github.com/onc-healthit/2022-LEAP-aligning-housing-and-healthcare/tree/main>
- Download the acleap-referral.zip file
- Move the zip to the local repository and extract the subdirectory structure.

6.2 Backend Service Application

- On a local device, create a directory to hold the repository
- Change path to the target local directory
- Login to GitHub
- Navigate to <https://github.com/onc-healthit/2022-LEAP-aligning-housing-and-healthcare/tree/main>
- Download the acleapbackendservice.zip file
- Move the zip to the local repository and extract the subdirectory structure.

6.3 Referral Handler Application

- On a local device, create a directory to hold the repository
- Change path to the target local directory
- Login to GitHub
- Navigate to <https://github.com/onc-healthit/2022-LEAP-aligning-housing-and-healthcare/tree/main>
- Download the acleapreferralhandler.zip file
- Move the zip to the local repository and extract the subdirectory structure.

7 Configuring and Uploading the Application Code

7.1 Establish WinSCP connections for the Azure Applications

- Gather the FTPS endpoint & credentials and configure for access
 - In Azure, Navigate to the application service.
 - In the left pane, expand Deployment and select Deployment Center
 - In the right pane, choose the FTPS credentials tab
 - Copy the FTPS endpoint, FTPS username and Password to notepad
 - In the left pane, choose Configuration
 - Turn on -
 - SCM Basic Auth Publishing
 - FTP Basic Auth Publishing (FTPS only)
- Repeat steps for each application
- Open Winscp

- Create New Site named Referral App
 - Protocol - FTP
 - Encryption - TLS/SSL Implicit Encryption
 - Host Name – Endpoint URL (port 990)
 - User Name – FTPS User Name
 - Password – Password
 - Save Site Info
 - Login
- Repeat steps for Backend Service and Referral Handler App

7.2 Referral (CBO) Application Deployment

- Locate the parameters which were documented during the App Registration in Azure (Section 4)
 - ClientID
 - TenantID
 - Redirect URL
 - Redirect URL for backend-service
- Open up command prompt in administrator mode
- Change directory to the location of the Referral (CBO) app repository
- Launch VSCode (“code .”)
- Edit the .env file (double click on the file)
- Add the documented parameters to the file
 - `REACT_APP_AZURE_ACTIVE_DIRECTORY_CLIENT_ID='xxxx-xxxx-xxxx-xxxx-xxxxxxx'`
 - `REACT_APP_AUTHORITY_LINK='https://login.microsoftonline.com/tenantid/'`
 - `REACT_APP_REDIRECT_URL='https://”referral app”.azurewebsites.net/'`
 - `REACT_APP_BACKEND_API_URL='https://”backend-service app”.azurewebsites.net'`
- Save the .env file (CTL s)
- Open the integrated terminal in VScode, by going to “View” > “Terminal” from the menu bar and make sure you are in correct directory and run the “**npm run build**” command for generating the build file. (If you are updating code and a build folder exists, delete the folder before running the new build.)
 - If the npm build within terminal does not work, open a command prompt in administrator mode and ensure you are running the correct node version.
 - Node -v (21.7.2 or current...if so, needs to be changed)
 - nvm install 16.20.2
 - nvm use 16.20.2
 - node -v (should now be 16.20.2)
 - npm run build
 - As an alternative, you can try using yarn
 - yarn run build
 - If it still fails, reinstall yarn
 - Yarn install
 - Yarn run build
 - Once the build completes, navigate to the build folder using file manager
 - Create a subfolder called acleap-referral
 - Move the static and assets subfolders into the acleap-referral folder

- Move manifest.json into the acleap-referral folder
 - In Azure, Stop the CBO service
- Open WinSCP
 - Login to the site for the Referral (CBO) App
 - In the left pane, navigate to the referral app build directory
 - In the right pane, navigate to /site/wwwroot/
 - Delete hostingstart.html if it exists
 - Copy contents of build folder to the /site/wwwroot/ folder in the right pane.
- In Azure, restart the CBO service
- After 5-10 minutes, the application should be active
- In application overview, click on the link for the default domain
 - If you get a popup blocked warning, change setting to allow popups from site
- If the login ID being used is a member of the permission group defined in section 4.4, the Referral Management Dashboard should launch properly

7.3 Backend Service Application Deployment

- Locate the parameters noted during app registrations
 - ClientID (from app registrations)
 - TenantID (from app registrations)
 - Client_Secret (from app registrations/Certificates & Secrets - Value)
 - Scope
 - Azure Portal,
 - Search for health data & ai services
 - FHIR service
 - Open the service
 - From Overview, copy URL for FHIR metadata endpoint
 - This will be pasted into Scope
 - without the “metadata” at the end
 - add /.default after removing metadata
 - Go to Access Control (IAM)
 - Role Assignments
 - Add / Role Assignment
 - FHIR Data Contributor
 - Select Members
 - Search for acleapbackend and select
 - Review and Assign (gives registrations access to FHIR service)
 - TOKEN_Url – URL containing tenant ID – example below
 - FHIR_SERVER_URL – Same as scope but without /.default
- Open up command prompt in administrator mode
- Change directory to the location of the Backend Service app repository
- Launch VSCode (“code .”)
- Edit the .env file (double click on the file)
- Add the documented parameters to the file
 - `TENANT_Id='xxxx-xxxx-xxxx-xxxx-xxxxxxx'` (from app registrations)
 - `CLIENT_Id='xxxx-xxxx-xxxx-xxxx-xxxxxxx'`
 - `CLIENT_SECRET='xxxx-xxxx-xxxx-xxxx-xxxxxxx'`
 - `SCOPE='https://acleap1mk622hds-ws-fhirtrn.fhir.azurehealthcareapis.com/.default'`

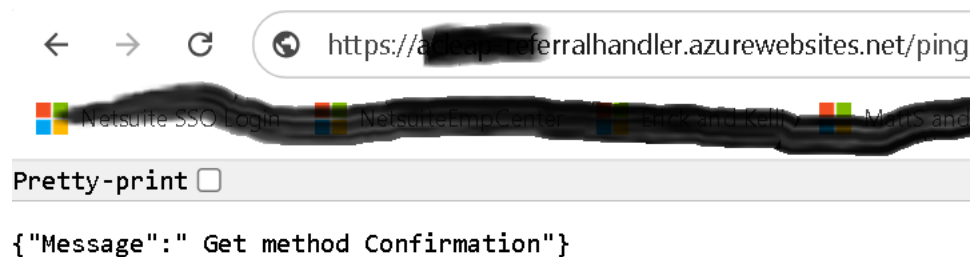
- `TOKEN_URL='https://login.microsoftonline.com/"tenantid"/oauth2/v2.0/token'`
- `FHIR_SERVER_URL=' https://acleaplmk622hdsws-fhirtrn.fhir.azurehealthcareapis.com'`
- Save the .env file (CTL s)
- **IMPORTANT – Do not restart the backend service in Azure after copying files but before using SSH to access site and running NPM install. The container will not come up and you will not be able to access it via SSH.**
- Open Winscp
 - Open Backend Service Site
 - In the left pane, navigate to the Backend Service app local repository
 - In the right pane, navigate to /site/wwwroot/
 - Copy hostingstart.html file from right pane to a temporary location on your local device
 - Delete hostingstart.html from right pane.
 - Copy all files except gitignore and readme from the left pane to the right pane
- Go to the Backend Service app in Azure Portal
 - Select SSH under Development Tools on the left side
 - `ls -a`
 - `npm install` (will install web files and directories)
- **If you accidentally restarted the service prior to running npm install, you may not be able to get the SSH connection working. To start over:**
 - **Connect back to backend service using Winscp**
 - **Delete the files you just copied from the right pane**
 - **Copy hostingstart.html back to the right pane**
 - **Restart the backend service**
 - **Follow process above to copy files to the backend service web site and SSH in.**
- In the left pane of the app under API, choose CORS
 - For Allowed Origins, put in *
 - Save
 - Stop and Start the App.
- Pasting the backend service app url into a browser window with /health appended to it confirms service is running



7.4 Referral Handler Application Deployment

- Locate the parameters
 - ClientID – (from Azure AD B2C | App registrations)
 - TenantID – (from Entra ID)
 - Client_Secret - (from Azure AD B2C | App registrations | Certificates & Secrets - Value)
 - ADScope – fhir url with /.default
 - Token_Url – URL containing tenant ID – example below
 - fhirServer_URL - fhir url
 - athenafhir_URL – from API configuration
 - athenaSubscription_KEY – from api service/subscriptions
- Open up command prompt in administrator mode
- Change directory to the location of the Referral Handler app repository
- Launch VSCode (“code .”)
- Right Click on Server and add file .env
- Edit the .env file (double click on the file)
- Add the documented parameters to the file (no single quotations this time)
 - `tenant_Id=xxxx-xxxx-xxxx-xxxx-xxxxxxxx`
 - `client_Id=xxxx-xxxx-xxxx-xxxx-xxxxxxxx`
 - `client_Secret=xxxx-xxxx-xxxx-xxxx-xxxxxxxx`
 - `ADscope=https://XXXX-fhirtrn.fhir.azurehealthcareapis.com/.default`
 - `Token_Url=https://login.microsoftonline.com/'tenantID'/oauth2/v2.0/token`
 - `fhirServer_URL=https://XXXXX-fhirtrn.fhir.azurehealthcareapis.com`
 - `athenafhir_URL=https://acleap-apimgmt.azure-api.net/XXresources`
 - `athenaSubscription_KEY= XXXXXX`
- Save the .env file (CTL s)
- Open getAthenaADToken.js
 - `Const apimUrl = 'https://acleap-apimgmt.azure-api.net/connectXX/token';`
(from api mgmt...add /token)
- **IMPORTANT – Do not restart the referral handler service in Azure after copying files but before using SSH to access site and running NPM install. The container may not come up and you may not be able to access it via SSH.**
- Open Winscp
 - Connect to Referral Handler App site
 - In the left pane, navigate to the Referral Handler app local repository
 - In the right pane, navigate to /site/wwwroot/
 - Delete hostingstart.html from right pane
 - Copy files from the Server directory in left pane to the right pane
 - Go to the referral Referral Handler app in Azure Portal
 - On left, go to SSH and hit Go
 - `Cd site/wwwroot`
 - `Ls -a`
 - `Npm install`
 - `Npm start`

- ***If you accidentally restarted the service prior to running npm install, you may not be able to get the SSH connection working. To start over:***
 - ***Connect back to referral handler service using Winscp***
 - ***In the right pane, delete the files you just copied***
 - ***Copy hostingstart.html back to the right pane***
 - ***Restart the backend service***
 - ***Follow process above to copy files to the backend service web site and SSH in.***
- Open the Referral Handler app in the Azure Portal
- In the left pane of the app service, go to CORS
 - Check Enable Access-Control-Allow-Credentials
 - Under allowed origins
 - Copy the acleap-apimgmt gateway URL (<https://acleap-apimgmt.azure-api.net>)
 - Save
 - Restart app service
 - Pasting the referral handler app url into a browser window with /ping appended to it confirms service is running.



7.5 Health Data Service

- In Azure, Open the Health Data Service
- Go to FHIR Service in the left pane
- Open FHIR Service
 - On left, open CORS
 - Place * in Origins
 - Place * in Headers
 - Methods – select all
 - Max Age – 600
 - Save
 - On left, go to Access Control
 - Role Assignments
 - Add Role
 - FHIR Data Contributor
 - Select Members –
 - acleapreferral(handler)
 - acleapCBO
 - Review and assign

8 Testing the Referral Workflow

- Trigger to generate test referral is sent to API Management Service IP
- Request needs to have the referral handler URL referenced in it
 - [https://\"referral app\".azurewebsites.net](https://\)
- You should be able to open referral and ad notes.

9 Manually Adding Resources to Referral App

Postman app is needed to manually add resources.

9.1 Adding Practitioners/Providers

- Open Postman for Windows
- Get the access token
 - Open Referral Handler code in VSCode and go to .env file
 - Copy/paste line 5 (token_Url) to notepad
 - Copy/paste line 2 (client_ID) to notepad
 - Copy/paste line 3 (client_Secret) to notepad
 - Copy/paste line 4 (ADscope) to notepad
 - Copy/paste line 6 (fhirServer_URL) to notepad
 - In postman, enter the token URL in the “Get” line.
 - Go to the “Body” tab.
 - Select x-www-form-urlencoded
 - Enter the values that were copied from the .env file to the key and value fields

- client_id
 - "client_ID"
- scope
 - "ADscope"
- client_secret
 - "client_Secret"
- grant_type
 - client_credentials (this is the literal value wording)
- Change the action from "Get" to "Post" and click "Send"
- In the response (bottom of screen under Body) , scroll down and find "access_token":
- Copy the access_token (without quotes) to notepad
- Open another tab in Postman
- Go to Authorizations tab
 - Choose drop down next to "Type" and select bearer token
 - Copy the access_token into the Token field
 - Paste the fhirServer_URL from notepad into the field next to "Get"
 - Add /Patient at the end of the URL
 - Hit Send
 - Change the Action from "Get" to "Post"
 - Go into the Body tab
 - Change type to raw
 - Select drop down next to text and change to JSON
 - Copy/Paste Provider Add script (sample below) into the field
 - In the "Post" URL, replace /Patient with /PractitionerRole
 - Hit Send

Single Provider Add

```
{
  "resourceType": "PractitionerRole",
  "id": "83ec4c33-66eb-4444-a0ee-60a706054248",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2024-02-20T19:08:43.355+00:00",
    "source": "#JkjOHpE7HwhYBlgJ"
  },
  "active": true,
  "practitioner": {
    "reference": "Practitioner/ch-practitioner-davis",
    "display": "Owen Davis"
  },
  "organization": {
    "reference": "Organization/mcc-org-chicago-house",
    "display": "Chicago House"
  },
  "code": [
    {
      "coding": [
```

```

        {
            "system": "http://nucc.org/provider-taxonomy",
            "code": "171M00000X",
            "display": "Case Manager"
        }
    ]
}
]
}

```

9.2 Adding Organizations

- Follow the same process that was used for Practitioners until you get to the Authorization Tab section
 - Go to Authorizations Tab
 - Choose drop down next to "Type" and select bearer token
 - Copy the access_token into the Token field
 - Paste the fhirServer_URL from notepad into the field next to "Get"
 - Add / PractitionerRole at the end of the URL
 - Hit Send
 - Change the Action from "Get" to "Post"
 - Go into the Body tab
 - Change type to raw
 - Select drop down next to text and change to JSON
 - Copy/Paste Organization Add script (samples below) into the field
 - In the "Post" URL, replace /PractitionerRole with /Organization
 - Hit Send
 - Change Action from "Post" to "Get" and hit send (one org at a time)
 - Body should now contain the Organization data

Raw JSON

```

{
  "resourceType": "Organization",
  "id": "ac-leap-heartland-alliance-health",
  "active": true,
  "name": "Heartland Alliance Health"
}

```

```

{
  "resourceType": "Organization",
  "id": "ac-leap-chicago-house",
  "active": true,
  "name": "Chicago House"
}

```

10 Manually Removing Resources to Referral App

Postman app is needed to manually remove resources.

10.1 Adding Practitioners/Providers

- Open Postman for Windows
- Get the access token
 - Open Referral Handler code in VSCode and go to .env file
 - Copy/paste line 5 (token_Url) to notepad
 - Copy/paste line 2 (client_ID) to notepad
 - Copy/paste line 3 (client_Secret) to notepad
 - Copy/paste line 4 (ADscope) to notepad
 - Copy/paste line 6 (fhirServer_URL) to notepad
 - In postman, enter the token URL in the “Get” line.
 - Go to the “Body” tab.
 - Select x-www-form-urlencoded
 - Enter the values that were copied from the .env file to the key and value fields
 - client_id
 - “client_ID”
 - scope
 - “ADscope”
 - client_secret
 - “client_Secret”
 - grant_type
 - client_credentials (this is the literal value wording)
 - Change the action from “Get” to “Post” and click “Send”
 - In the response (bottom of screen under Body) , scroll down and find “access_token”:
 - Copy the access_token (without quotes) to notepad
 - Open another tab in Postman
 - Go to Authorizations Tab
 - Choose drop down next to “Type” and select bearer token
 - Copy the access_token into the Token field
 - Paste the fhirServer_URL from notepad into the field next to “Get”
 - Add /PractitionerRole at the end of the URL
 - Hit Send
 - Find the ID for the provider you are trying to delete
 - Append the ID after /PractitionerRole/
 - With the action still set to Get, select Send
 - In the body, verify that you are working with the right provider
 - If the provider is correct, ensure that the URL still has the /PractitionerRole/”ID” set correctly, change the action from Get to Delete and select Send **(this will delete the provider you are working with)**
 - To verify, Change the Action from “Delete” to “Get” and select Send
 - The body tab should be blank.

10.2 Removing Referrals

- Same process as 10.1 up to the “Go to Authorizations Tab” section
 - Go to Authorizations Tab
 - Choose drop down next to “Type” and select bearer token
 - Copy the access_token into the Token field
 - Paste the fhirServer_URL from notepad into the field next to “Get”
 - Add /Task at the end of the URL
 - Hit Send
 - List of tasks should be displayed in the Body tab
 - Find the “id” for the referral you want to delete
 - Append the ID after /Task/
 - With the action still set to Get, select Send
 - In the body, verify that you are working with the right referral
 - If the referral is correct, ensure that the URL still has the /Task/“ID” set correctly, change the action from Get to Delete and select Send **(this will delete the referral you are working with)**
 - To verify, Change the Action from “Delete” to “Get” and select Send
 - The body tab should be blank.

11 Updating Applications in Azure

Postman app is needed to manually remove resources.

11.1 Referral (CBO) Application

- Change path to the target local directory
- Create a backup of the current file structure
- Login to GitHub
- Navigate to <https://github.com/chronic-care/acleap-referral.git>
- Change branch to “develop”
- Select the Code button and choose “Download Zip”
- Move the zip tot the local repository and extract the subdirectory structure.
- Copy .env file from the backup area
- Launch VSCode (“code .”)
- Verify that the .env file has the correct parameter entries
- Open terminal from within VSCode
- Cd acleap-referral
- Node -v (verify v16.20.2)
- Npm run build
 - If that doesn’t work, try yarn run build
 - If that does not work, run yarn to download modules
 - (if build folder exists in the referral app code directory, delete it)
 - Now try “yarn run build” again
- In command prompt, navigate to the build folder in the referral (CBO) app directory
- Under build, create a subfolder caller acleap-referral
- Move the assets and static folders into the acleap-referral folder
- Move the manifest.json file into the acleap-referral folder
- In Azure, stop the Referral (CBO) application
- Open WinSCP and connect to site
- Copy all contents of build folder to the /site/wwwroot/ folder

- In Azure portal, restart the referral (CBO) application.

11.2 Referral Handler Application

- Change path to the target local directory
- Create a backup of the current file structure
- Login to GitHub
- Navigate to <https://github.com/chronic-care/ACLeapReferralHandler> "
- Select the Code button and choose "Download Zip"
- Move the zip tot the local repository and extract the subdirectory structure.
- Copy the .env file from the backup directory to the current code directory
- Launch VSCode ("code .")
- Open the .env file and verify the parameters.
- Open Winscp
 - Connect to Referral Handler App site
 - In the left pane, navigate to the Referral Handler app local repository
 - In the right pane, navigate to /site/wwwroot/
 - Copy the hostingstart.html file to a temp local directory. (in case ssh does not work)
 - Delete hostingstart.html from right pane
 - Copy files from the Server directory in left pane to the right pane
 - Go to the referral Referral Handler app in Azure Portal
 - On left, go to SSH and hit Go
 - Cd site/wwwroot
 - Ls -a
 - Npm install
 - Npm start
- In Azure, restart the Referral Handler application

11.3 Backend Service Application

- Change path to the target local directory
- Create a backup of the current file structure
- Login to GitHub
- Navigate to <https://github.com/chronic-care/AcleapBackendService>
- Select the Code button and choose "Download Zip"
- Move the zip tot the local repository and extract the subdirectory structure.
- Copy the .env file from the backup copy of the app to the current repository
- Open WinSCP and connect to site
- To copy specific files
 - In the left pane, navigate to the Backend Service app local repository
 - In the right pane, navigate to /site/wwwroot/
 - Copy files that have been updated
- To refresh the entire app
 - In the left pane, navigate to the Backend Service app local repository
 - In the right pane, navigate to /site/wwwroot/
 - Copy the hostingstart.html file to a temp local directory. (in case ssh does not work)
 - Delete hostingstart.html from right pane.
 - Copy all files except gitignore and readme from the left pane to the right pane
 - Go to the Backend Service app in Azure Portal
 - Select SSH under Development Tools on the left side

- ls -a
 - npm install (will install web files and directories)
- In Azure, restart the Backend Service application