1 Introduction

This document provides a detailed walkthrough of steps required to deploy Virtual Health. The deployment guide explains the deployment of following components.

Component Name	Component Description	Component Type
HealthCare.Portal	Virtual Health Web App	Web App
SharePoint assets	SharePoint artifacts	SharePoint Site
VHCBot	Virtual Health BOT for Scheduling	Web App
HealthCare.Web	Virtual Health Angular Web App	Web App

2 System Requirements

2.1 Office 365 Plan Requirements

It is recommended to use Office 365 Enterprise E3 and above

E3 Plan details are available here:
https://products.office.com/en-us/business/office-365-enterprise-e3-business-software

Other plans:
http://office.microsoft.com/en-us/business/compare-all-office-365-for-business-plans-FX104051403.aspx

2.2 Azure Subscription

The Virtual Health Templates requires an Azure subscription to host the following services

- Website/Web App
- Application Insight
- Key vault
- Azure table

2.2.1 Minimum Azure Web App Configuration

The Virtual Health web apps need at least a Standard configuration to cater the needs of pilot. However, it can be scaled out as per the application need.

<table>
<tr><th>Configuration</th><th>Details</th></tr>
<tr><td>Mode – Standard

Instances – A single instance in Shared or Standard mode already benefits from high availability, but you can provide even greater throughput and fault tolerance by running additional web site instances. In Standard mode, you can choose from 1 through 10 instances, and if you enable the Auto scale feature, you can set the minimum and maximum number of virtual machines to be used for automatic scaling.

http://www.windowsazure.com/en-us/documentation/articles/web-sites-scale/</td><td>Web Sites Standard (Promotional Pricing): The Standard tier offers multiple instance sizes as well as scaling to meet changing capacity needs. Prices for Standard are as follows:

<table>
<tr><th>SIZE</th><th>CPU CORES</th><th>MEMORY</th><th>PRICE PER HOUR</th></tr>
<tr><td>Small</td><td>1</td><td>1.75 GB</td><td>$0.10
(~$74 / month)</td></tr>
<tr><td>Medium</td><td>2</td><td>3.5 GB</td><td>$0.20
(~$149 / month)</td></tr>
<tr><td>Large</td><td>4</td><td>7 GB</td><td>$0.40
(~$298 / month)</td></tr>
</table>

Note: Refer to the below link to know more about the pricing models: http://www.windowsazure.com/en-us/pricing/details/web-sites/</td></tr>
</table>

2.2.2 Software Requirements

Since the services will be deployed in Azure PaaS, there is no separate software requirements

3 Prerequisites

The following prerequisites are important for the virtual health application

Office 365	Details
Plan	Purchase Office 365 Enterprise E3 plan: https://products.office.com/en-us/business/office-365-enterprise-e3-business-software Other plans: http://office.microsoft.com/en-us/business/compare-all-office-365-for-business-plans-FX104051403.aspx
Domain(Optional)	Domain for Office 365 *This is optional for Virtual Health Deployment*
Site Collection	Provision a site collection for Virtual Health. Preferably Publishing site

Azure	Details
Azure Subscription	Azure subscriptions will host following services • Website/Web App • Application Insight • Key vault (*Optional)* • Azure table
SSL certificate	SSL certificates are required for azure web sites and key vaults. It is recommended to have two CA issues SSL certificates for the domain CA issued certificates are required for Trusted Application Endpoint configuration and deployment **Optional if you are not using KeyVault and not deploying the trusted application endpoint as Cloud Service*

Active Directory Integration	Setup and synchronize existing Organization Active Directory on O365 portal http://technet.microsoft.com/en-us/library/hh967642
Domain (Optional)	Domain for azure website
Azure Apps	Provision Azure Web App for the Virtual Office Solution
Application Insights	Provision Application insights for the Virtual Office Solution
Key Vault	Provision a key vault for the Virtual Office Solution. This is optional **Optional if you are not using Key Vault*
User Account	User should have access to provision and configure services in the Azure PaaS and should be site collection Administrator • Azure Subscription Admin or similar Role • Office 365 Site Collection Administrator
Trusted Application Endpoint	This application must be deployed as azure cloud/azure app service before the deployment of Virtual Health. Refer to link Trusted Application API

☐ Before you proceed for Virtual Health deployment, you must deploy the **Trusted Application Endpoint**

Trusted App Sample Code Download Link	Usages

https://github.com/OfficeDev/skype-docs/tree/master/Skype/Trusted-Application-API/samples/AnonMeetingJoinSamples	Download the AnonMeetingJoinSamples and deploy the sample as azure cloud service or azure app service
	Note down the https endpoint of the Cloud Service e.g. https://yourclouservice.com
	This will be used in Web.Config of Virtual Health Portal (HealthCare.Portal) for key *TrustedApi*
	`<!--Trusted API enpoint-->`
	`<add key="TrustedApi" value="HTTPS_URI_TRUSTED_APP" />`

3.1 Certificates Required for Deployment

Certificates required for deployment are given in the below table

Certificate Type	Application	Purpose
Self-signed or CA issued* **Only required if Trusted Application is going to use certificated based authentication (OAuth flow) else client and secret based OAuth flow should work fine*	Trusted Application Endpoints	This certificate will be used to setup the OAuth with Azure AD application
CA issued* **To access the cloud service, you need to host it over https. Hence you need a CA issued certificate* *Certificate is not required if Trusted Application Endpoint is deployed as Azure App service*	Trusted Application – Cloud Service	This certificate will be required to configure the https endpoint for Cloud Service
Self-Sign or CA Issues ***Optional if Key Vault is not used for Virtual Health*	Key Vault Application	This certificate will be used to setup Key Vault application access. This certificate will be used in section 4.5

4 First time configuration

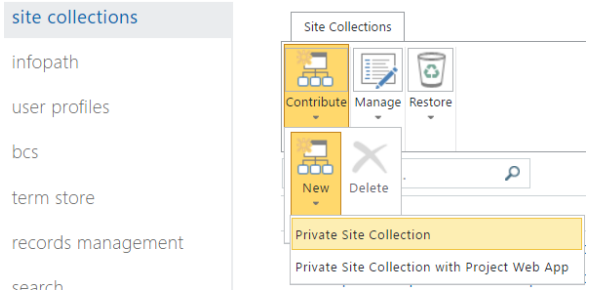
This section will provide steps to configure any new environment for the first time. This step is not required once the new environment is setup.

> Open a notepad or xml notepad to note down the configuration values as you go through this section. These configurations will be used during the deployment of the Web Apps

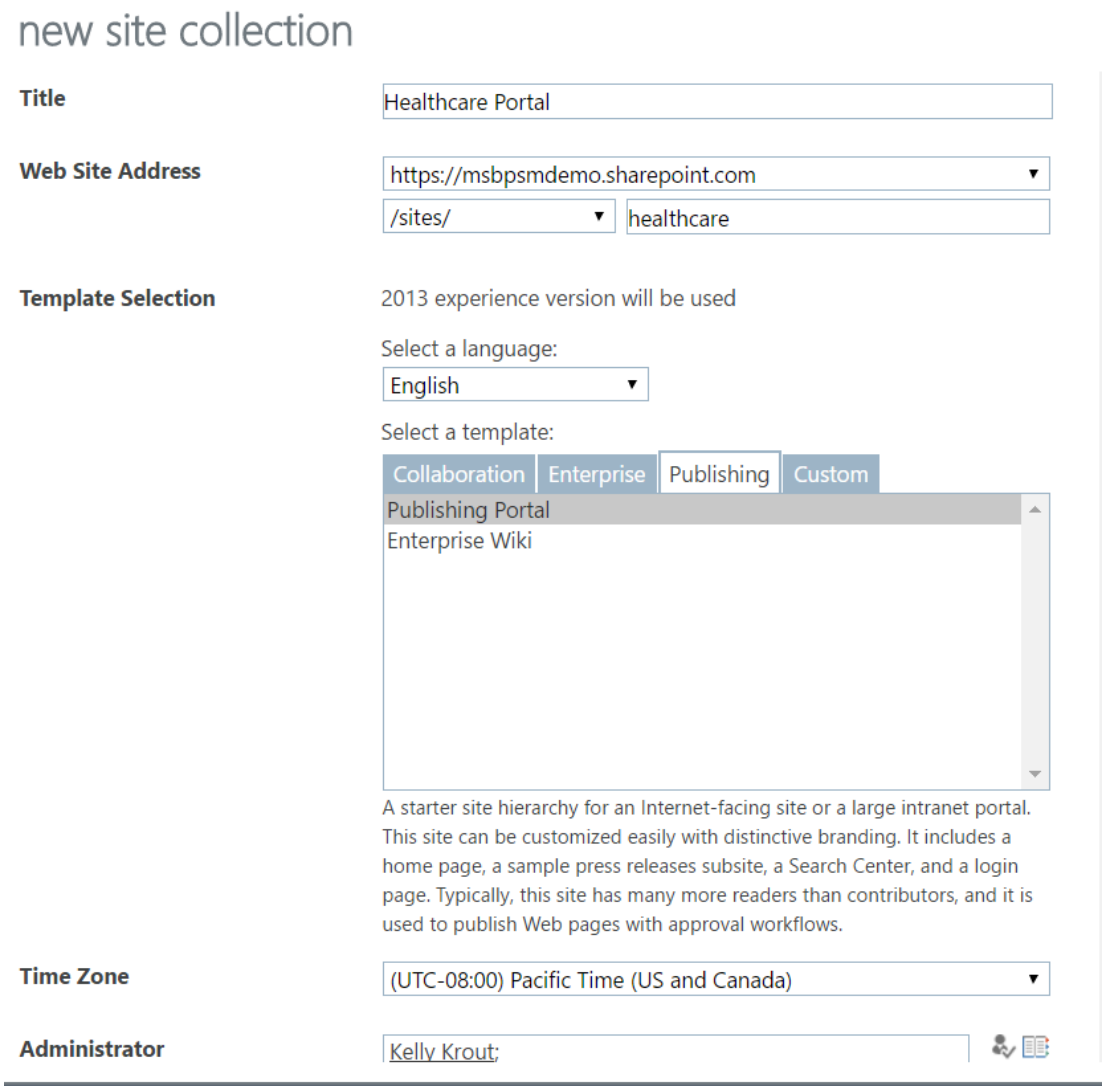
4.1 SharePoint Site Collection Provisioning

This section describes the steps required to provision a SharePoint site collection

A. Provision SharePoint site collection
   a. Sign in to the Office 365 admin center with your SharePoint Online admin user name and password
   b. Go to Admin > SharePoint
   c. Click on site collection tab
   d. Contribute → New → Private Site Collection



   e. Click on Private Site Collection
   f. Fill in the details as shown below



g. Click Ok
h. Note down the site collection URL in below format

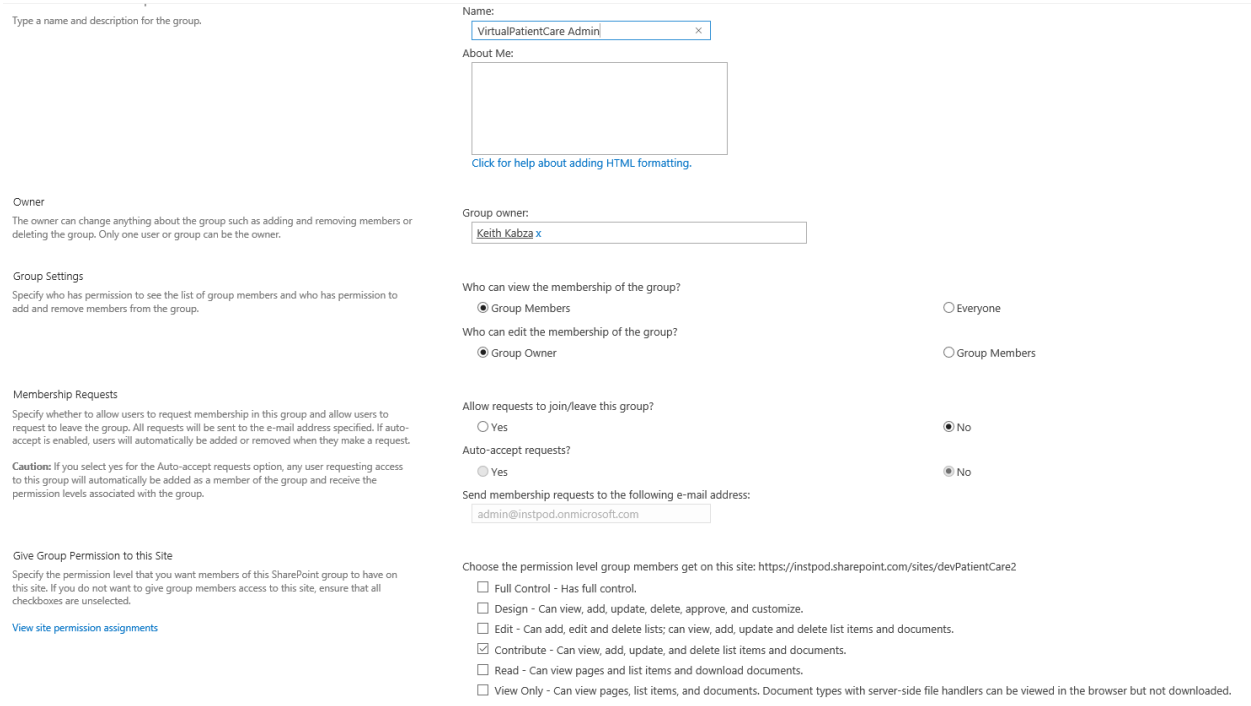
```
<add key="SharepointSite" value="SITE_COLLECTION_URL" />
```

4.2 SharePoint Configurations

Create a SharePoint Group and Add the people to the group who will have access to the settings page of the Virtual Health solution

- Open the site collection
- Go to site settings → Peoples and Group
- Click on More
- Click on New

- Fill in the details like Name as "`VirtualPatientCare Admin`" and select group permission as contribute



- Click Save
- Note down the SharePoint Group Name in below format

```
<add key="SharepointAdminGroup" value="SHAREPOINT_GROUP_NAME" />
```

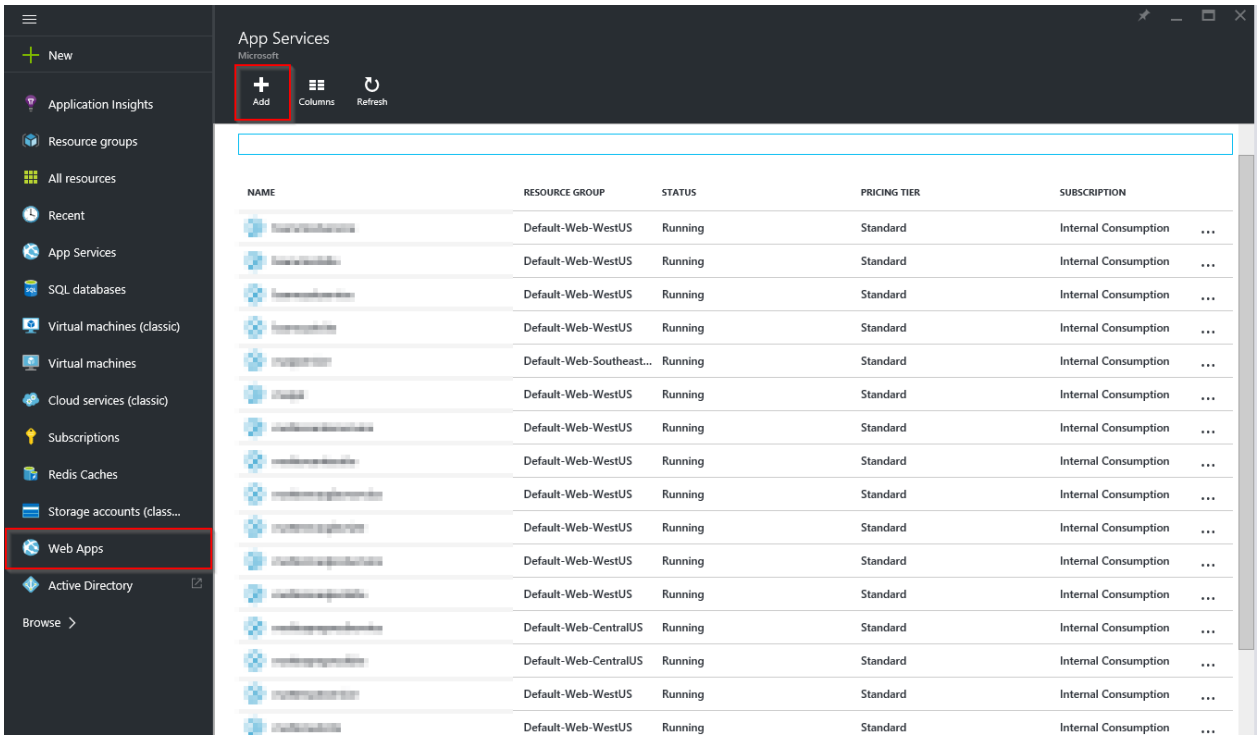
4.3 Azure Web Apps Provisioning

Virtual Health solution has following web apps

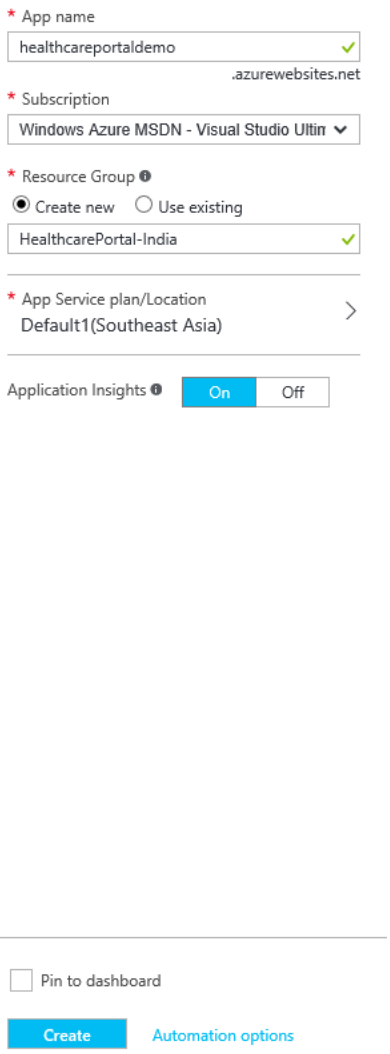
- HealthCare.Portal
- VHCBot
- HealthCare.Web

All the above web apps can be provisioned using below steps (steps only explaining for HealthCare.Portal web app)

A. Steps to provision an azure website
   1. Login to Azure management portal
   2. Select Web Apps and click on Add button



3. Enter name for web app and appropriate subscription Resource group and App Service plan as shown below



4. Click on create button at the bottom of the panel
   Note: *You should select the App Service Location closest to your users. This will help in reducing network latency and potentially provide better experience for users. App Service Location (in above example Southeast Asia) will be used to automatically create other services in the same location (in above example it will be Southeast Asia) so that all required objects are co-located.*

B. Steps to configure Application Insights
   a. While creating the website if you mark the option "Applications Insights" to **ON,** Application Insight will be provisioned
   b. Note down the application insight key

C. Website – Scaling
   a. If you want to change the scaling of the website, you can scale as per pre-requisite section

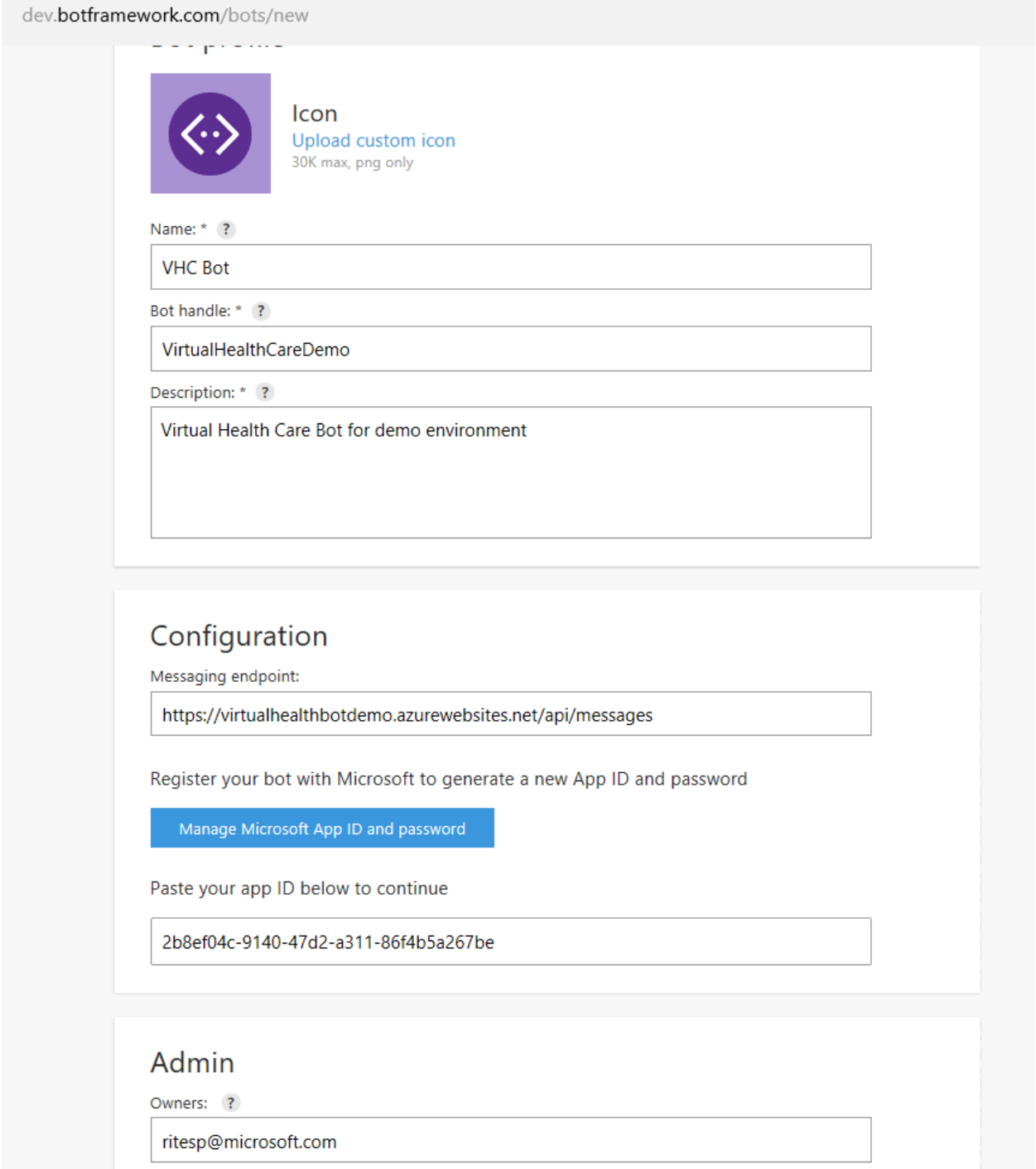
Note down the following values from this section

```
<add key="ida:HealthCarePortal" value="HEALTH_CARE_PORTAL_URL />

<add key="iKey" value="APP_INSIGHT_KEY_HEALTH_CARE_PORTAL" />

BOT_WEBSITE_URL = https://xxxx.azurewebsites.net [It will be used in BOT
Configuration Section]
```

4.4 Bot Configuration

Register a Bot

- Go to https://dev.botframework.com
- Click on Register a bot
- Add the following details
    - Name: display name
    - Bot handle: unique ID, not used elsewhere
    - Messaging endpoint: HTTPS endpoint used by the bot framework; if Azure Bot Web App deployment is on ***https://x.azurewebsites.net*** then this will be https://x.azurewebsites.net/api/messages



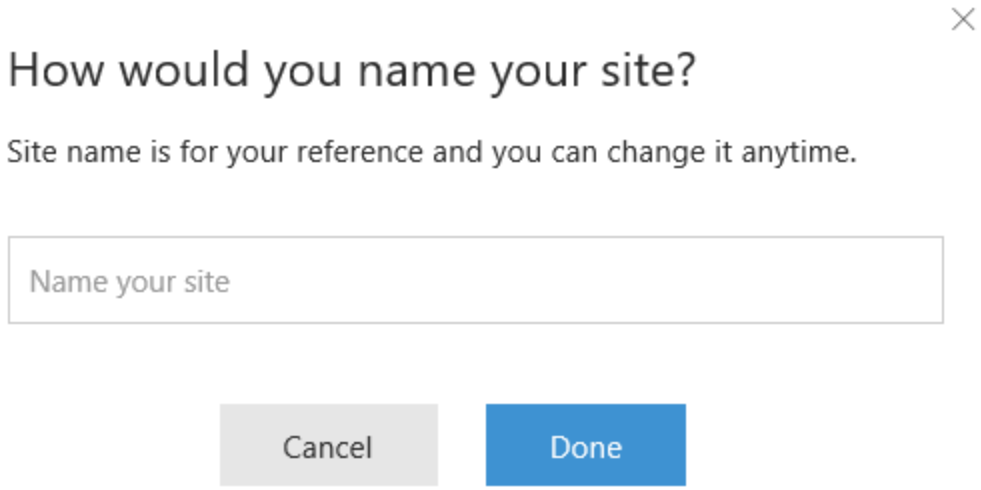
- Click “Create Microsoft App ID and Password”
- It will generate App Id and password and make a note of it, and you need to update same in web.config file of Bot Project
- Click on Register to register the bot
- Note down the configurations values in below format

```
<add key="BotId" value="BOT_ID" />

<add key="MicrosoftAppId" value="BOT_APP_ID" />

<add key="MicrosoftAppPassword" value="BOT_APP_PASSWORD" />
```

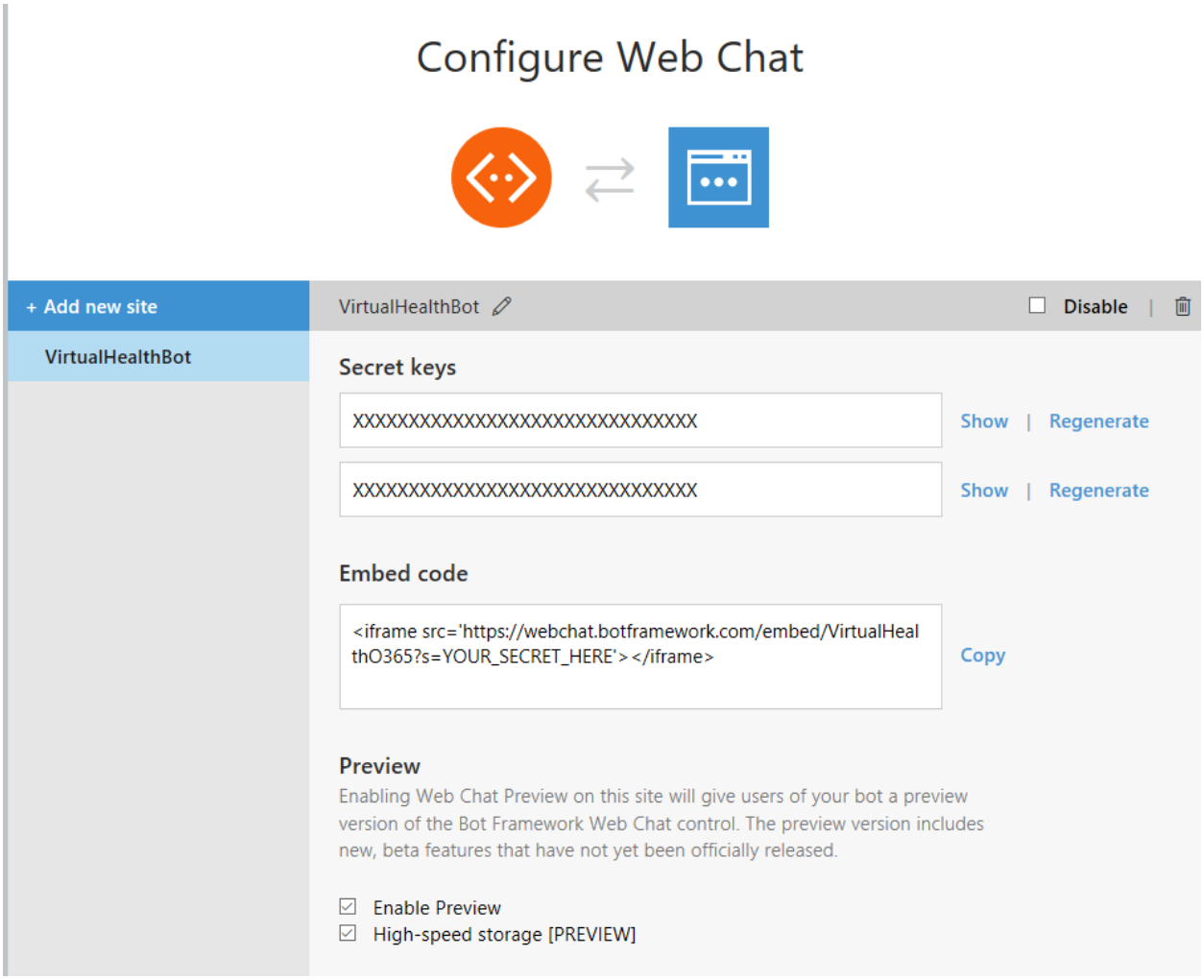
To get the Bot embed code, follow the below steps (Execute this step after execution of 5.4)

- Go to https://dev.botframework.com after registering the Bot
- Click on My Bots
- Click on the Bot created for the Virtual Health
- Click on Edit link available for Web Chat under Channel section

- Click on **Add New Site**
- Type the name like "VirtualHealthBot"



- Click Done, you will be redirected to Configure Web Chat page



- Copy the embed URL like https://webchat.botframework.com/embed/VirtualHealthO365?s=YOUR_SECRET_HERE
- Click on the Show secret and copy it and Replace the YOUR_SECRET_HERE with the secret
- Copy the embed URL as it will be required to be updated in Web.Config file of HealthCare.Portal web project.
  Note down the configuration value in below format

```
<add key="botUrlEmbed" value="BOT_EMBED_URL" />
```

4.5 Key Vault Provisioning

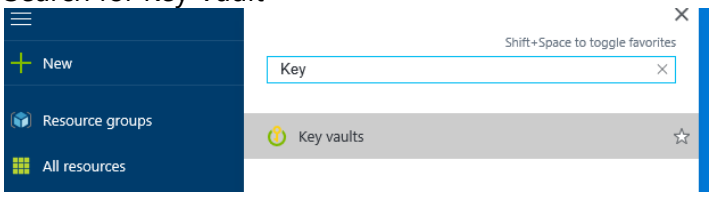
Execute this step only if you are going to use Key Vault. If you are using Key Vault, make sure the web.config of Virtual Health Web App has `<add key="IsKeyVaultEnabled" value="true" />` ***otherwise keep this value equal to false***

Key Vault Provisioning and Configuration

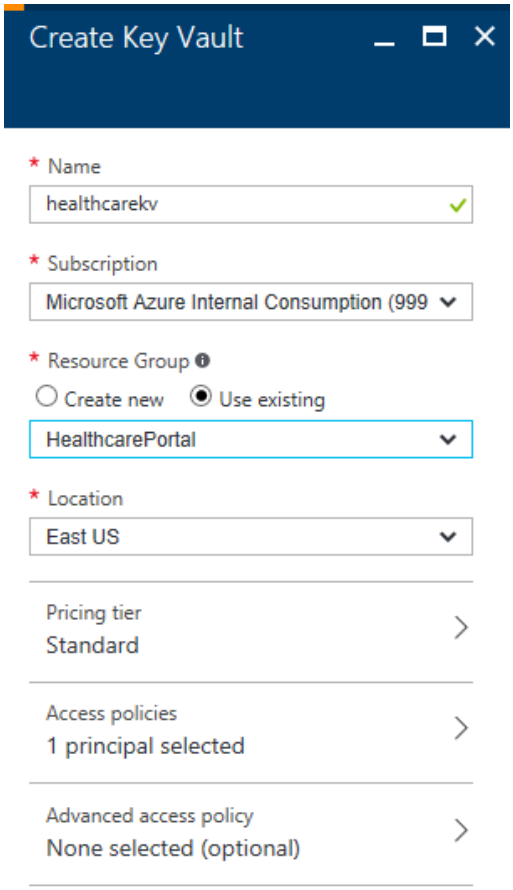
Note:* User should have access to

- Create an active directory application using PowerShell
- Assign the service principal to an azure active directory application.

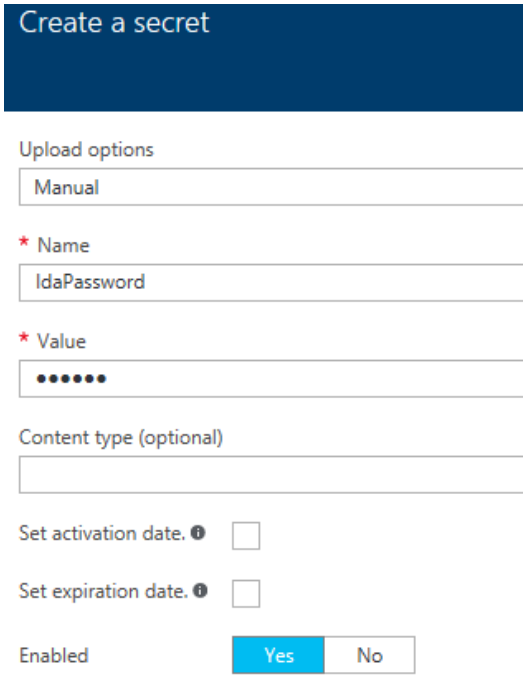
A. Key Vault Provisioning
   a. Login to Azure management portal
   b. Click on New (+) on left navigation Panel
   c. Search for Key Vault



   d. Click on Key Vaults
   e. Click on Add
   f. Fill in the details as shown below



g. Click Create
h. Once it is provisioned, open the Key Vault
i. Click on Secrets
j. Click Add and fill the values like below



k. Click Create
l. Add another secret with Name "SpoPassword", "EncryptionKey" and "EncryptionSalt". The values of these secrets will be like
SpoPassword: Password of a spo user
IdaPassword: password of AAD user (most of the case it is same as spo user)
EncryptionKey: Generate a new GUID
EncryptionSalt: Generate a new GUID

> Please generate new Guid for EncryptionKey and EncryptionSalt

m. Note down the Key Vault Base URL in below format

```
<add key="KeyVaultBaseUrl" value="KEY_VAULT_BASE_URL"/>
```

B. Configure Azure AD application for Key Vault and associate certificate
a. Get the certificate for the Key Vault or Create a self-signed certificate using the link https://technet.microsoft.com/itpro/powershell/windows/pki/new-selfsignedcertificate
Open the PowerShell command as Administrator
b. Run the following PowerShell after updating the yellow highlighted below
It does create following items
- Creates the AD application with the certificate
- Create service principal
- Assign reader role to the service principal

You need to replace the below yellow highlighted text with actual values for your environment.

Add-AzureRmAccount

```
PS C:\WINDOWS\system32> Add-AzureRmAccount
```

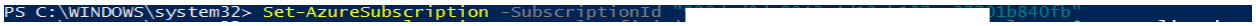
//The account that has privileges to create and assign service principal in the azure AD

$cert = New-SelfSignedCertificate -CertStoreLocation "cert:\LocalMachine\My" -Subject "CN=o365virtualhealth" -KeySpec KeyExchange

$keyValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

$app = New-AzureRmADApplication -DisplayName "virtualhealthKv" -HomePage "https://virtualhealtho365" -IdentifierUris "https://virtualhealtho365/virtualhealth" -CertValue $keyValue -EndDate $cert.NotAfter -StartDate $cert.NotBefore

//If you have multiple subscription make sure, you use following command



New-AzureRmADServicePrincipal -ApplicationId $app.ApplicationId

New-AzureRmRoleAssignment -RoleDefinitionName Reader -ServicePrincipalName $app.ApplicationId

$app

$cert.ThumbPrint

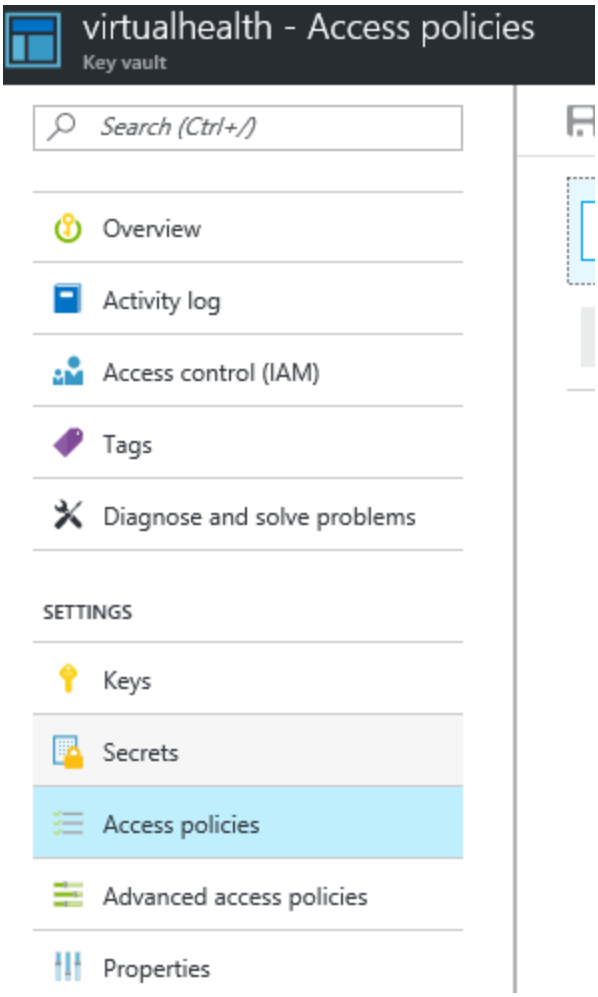
Refer to this link for more details https://docs.microsoft.com/en-in/azure/azure-resource-manager/resource-group-authenticate-service-principal

c. Note down the thump print and application Id as it will be used in web.config of the HealthCare.Portal application

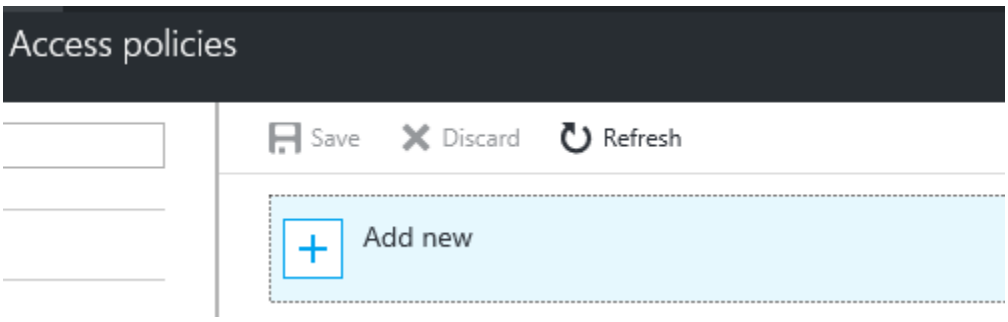
```
<add key="ClientId" value="AZURE_AD_APPLICATION_ID" />

<add key="Thumbprint" value="THUMB_PRINT_CERTIFICATE" />
```

Add Access Policy to Key Vault

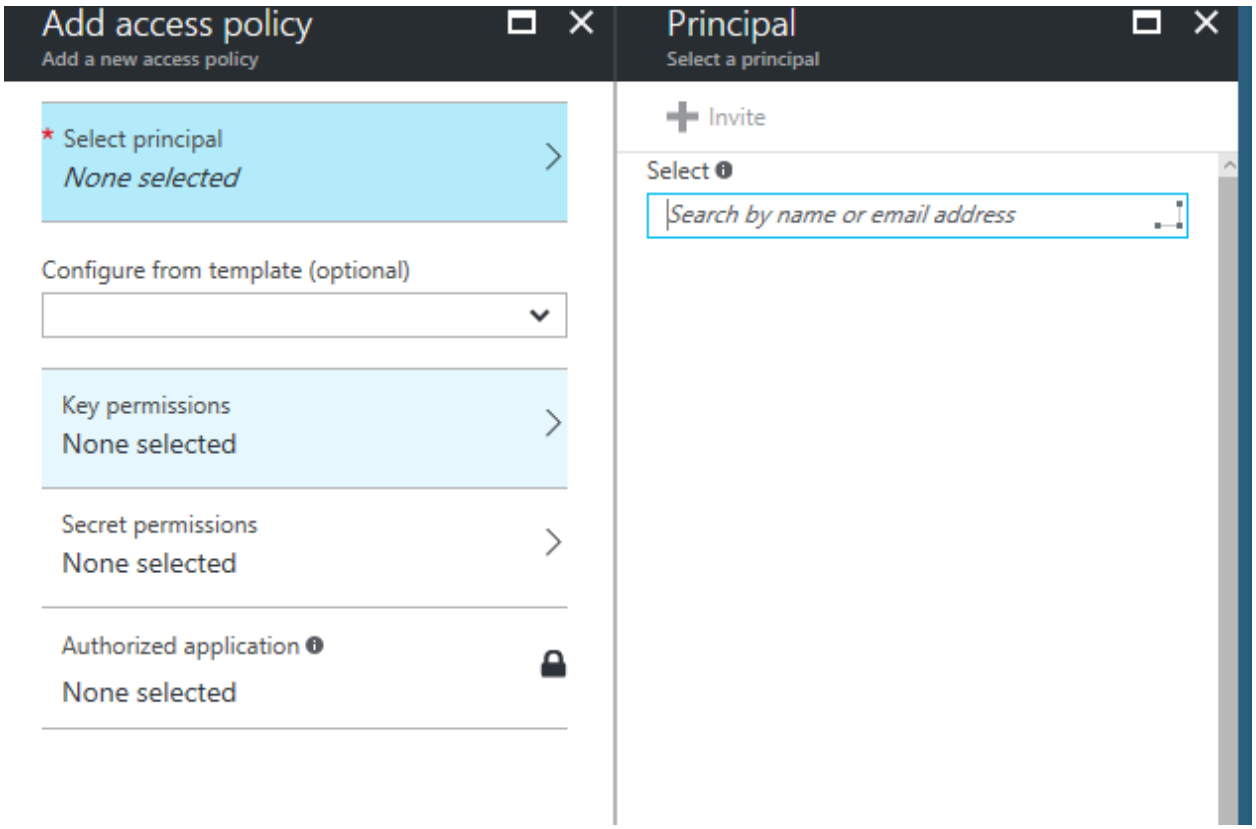
1. Go to the Azure Portal
2. Browse key Vault created earlier
3. Go to Access Policies



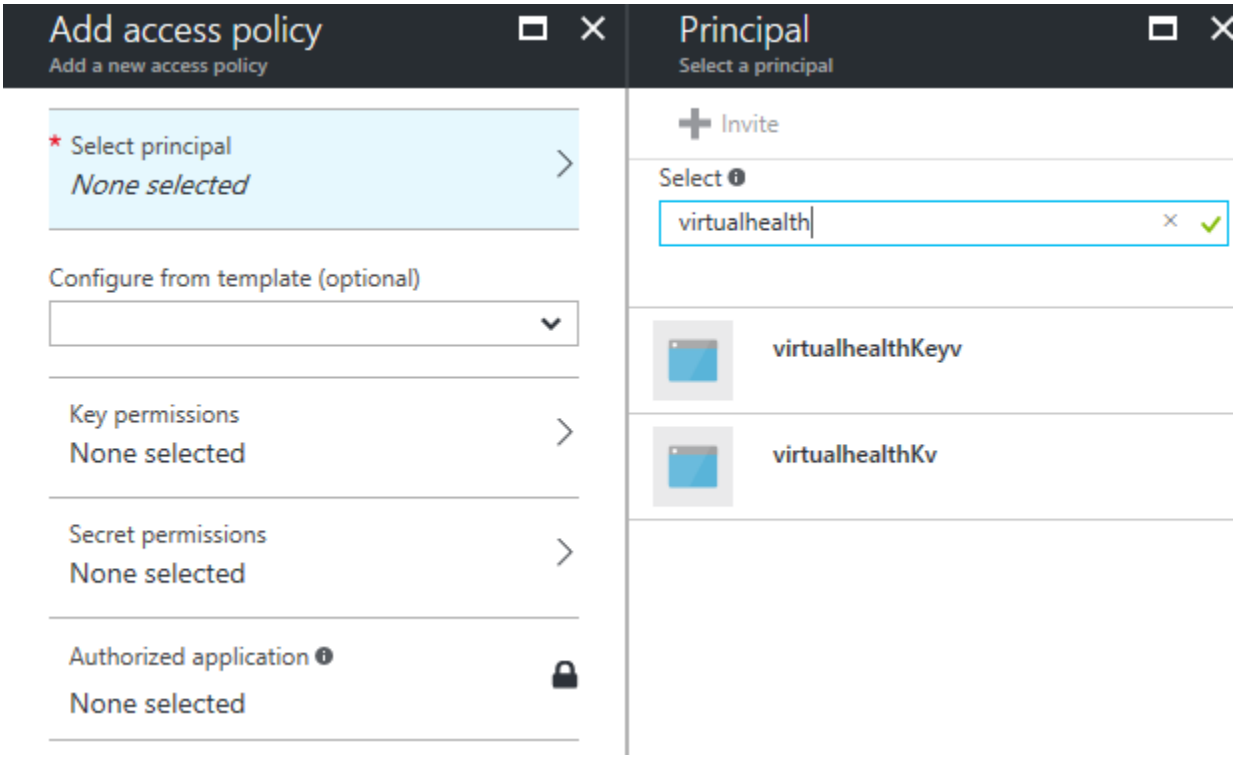
4. Click on Add New



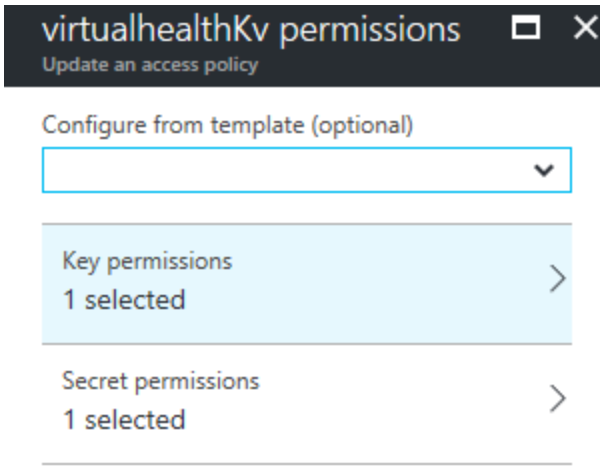
5. Click on Service Principal



6. Search the principal that was created earlier like virtualhealth



7. Select the one which was created in earlier step
8. Select the permission for Key and Secret (at least get permission should be assigned)

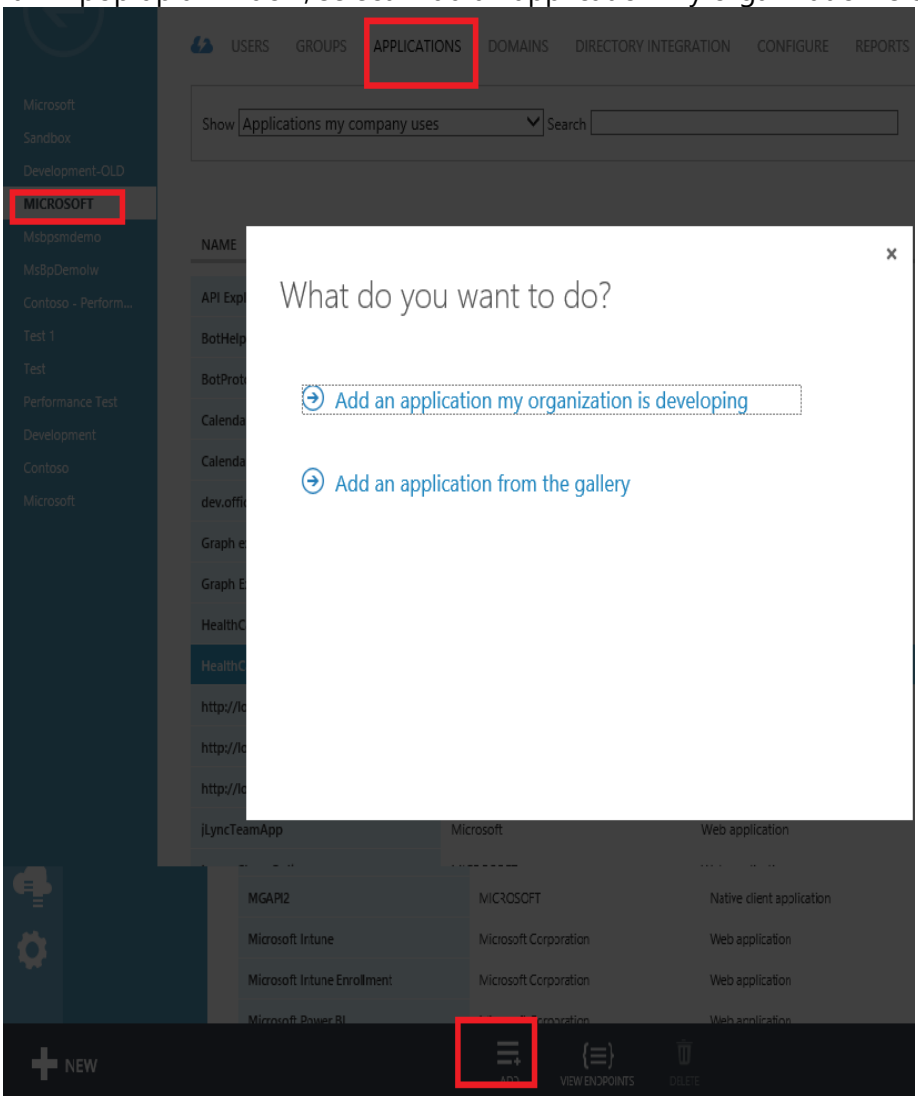

virtualhealthKv permissions
Update an access policy
Configure from template (optional)
Key permissions
1 selected
Secret permissions
1 selected

9. Click Save

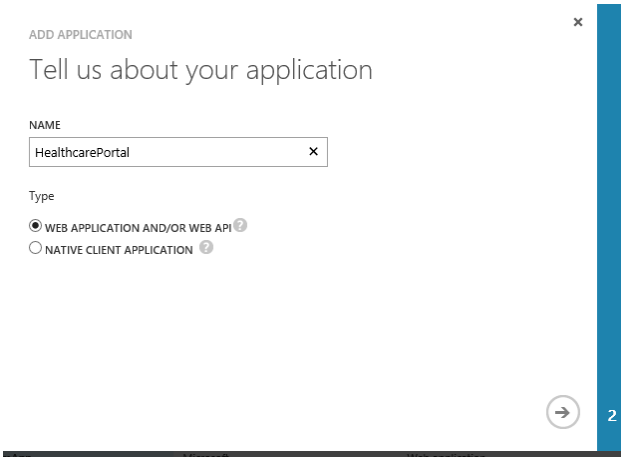
4.6 Azure AD application Configuration

Provision Azure AD applications

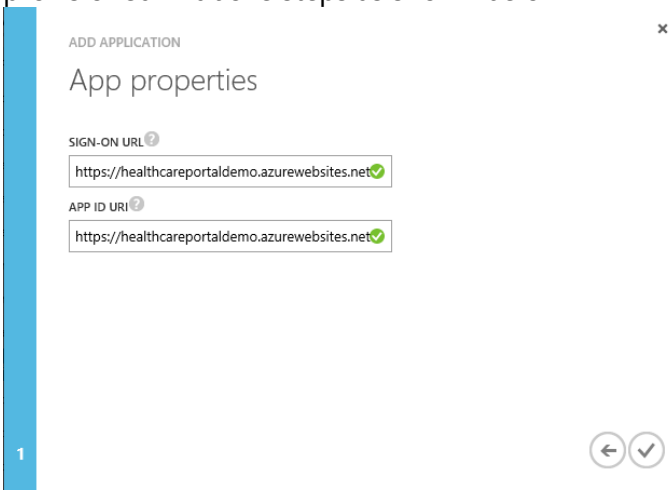
1. Login to Classic Azure Management Portal with Azure Admin Account
2. Click on the Active Directory link on the left menu
3. Select the Active Directory
4. Click on Applications tab
5. Then click on Add link at bottom
6. It will pop up a window, select "Add an application my organization is developing"



A. Provision Azure Application for HealthCare Portal
   a. Select "Web Application AND/OR Web API" option
   b. Provide name "HealthcarePortal"

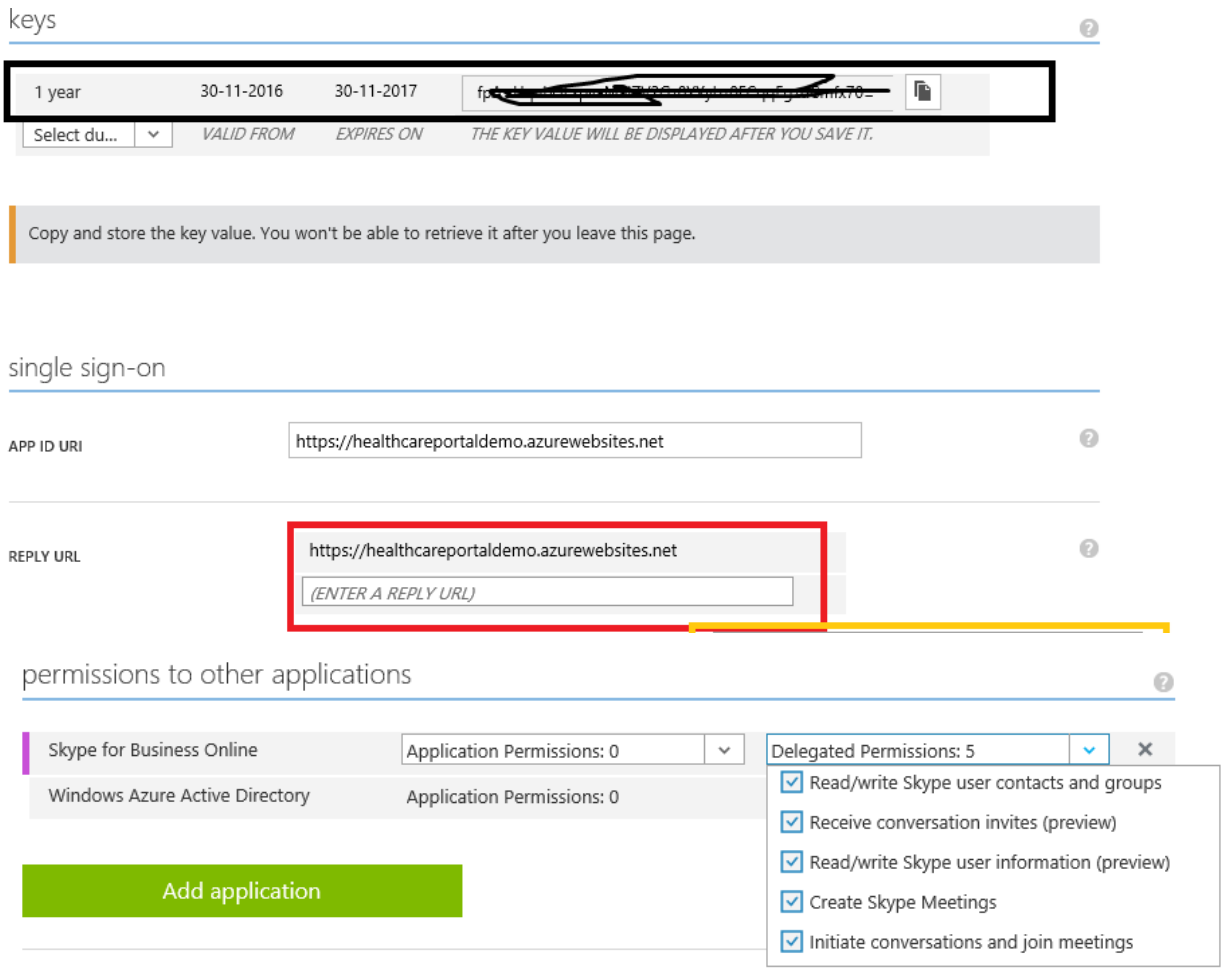


c. Click next
d. Enter sign-ON and APP ID URL as valid URL e.g. Azure Portal Web App provisioned in above steps as shown below



e. Click Ok
f. Click on the created application
g. Then navigate to Configure tab
h. Go to key's section, add a key or select duration and save the application settings by clicking save
i. It will generate a secret against key, **make a note of it**
j. **Make a note of client Id**
k. Now go to single sign-on section and add reply URL as per application URL i.e. HealthCare.Portal URL
l. Add reply URL for the Health Care Angular App as per application URL i.e. **HealthCare.Web** URL (This is required only if you are planning to host HealthCare.Web Angular app)
m. Also add the following reply URL for Angular App **https://hostname[name of angular web app].azurewebsites.net/token.html**
n. Go to permissions to other applications
o. Click on Add application

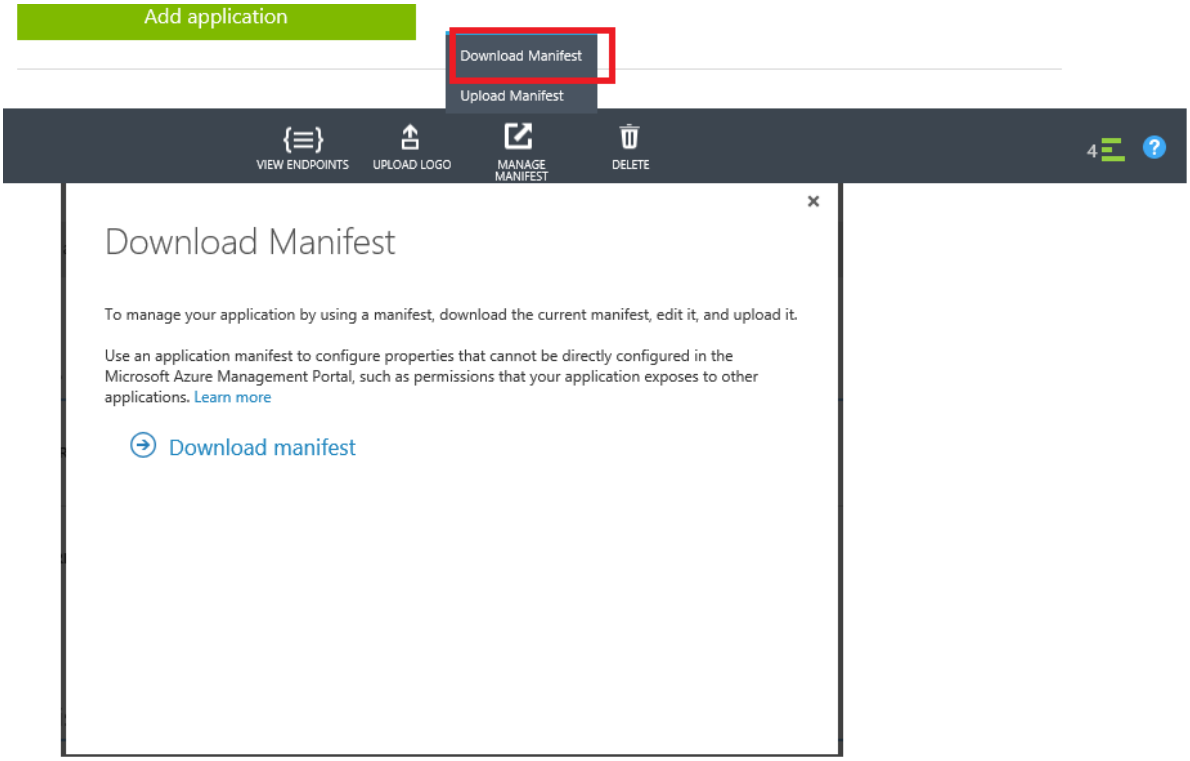
p. Select Skype for business online and click ok
q. Select the delegated permissions as shown in the below picture



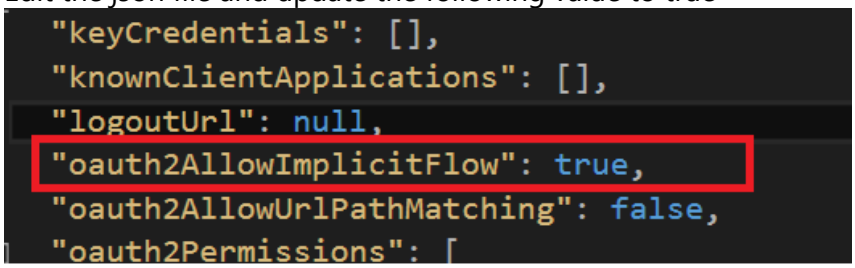
r. Click on save to save the application settings
s. **Note down the application secret from keys sections**

```
<add key="ida:ClientId" value="CLIENT_ID_AZURE_AD_APP_HEALTH_CARE_PORTAL" />

<add key="ida:ClientSecret" value="CLIENT_PWD_AZURE_AD_APP_HEALTH_CARE_PORTAL " />
```

B. Configure OAuth Flow
a. Open the Azure AD application created in step A
b. Go to Configure tab
c. At the bottom, click on the Manage Manifest
d. Click on Download Manifest



e. Edit the json file and update the following value to true



f. Save the file

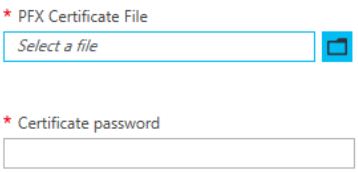
g. Upload the manifest file

4.7 Azure Web App Configurations

This step is optional if you are not using Key Vault

The following configuration is required on the HealthCare.Portal Web App

A. Add the certificate to Web App
   a. Click on App Services available in Left Navigation of Azure Portal
   b. Click on the Web App
   c. Then Click on SSL Certificates
   d. Then click upload certificate
      Note: If you are using self-signed certificate then this certificate is the same certificate which was created during the Key Vault provisioning. Export the *.pfx file with password. If you are using CA certificate, then you should be having *.pfx file available to you.
   e. Select *.pfx file and enter the password for the PFX file



   f. Click Upload

B. Add the Key to Azure Web App
   a. To load the certificate, add the following entry in azure web app
   b. Click on the Application Settings
   c. Scroll down and look for App Settings section
   d. Enter WEBSITE_LOAD_CERTIFICATES as key and * as value
   e. Then save the setting.

4.8 Other Configurations

This section will list down other configurations elements that needs to be captured before deploying the Web Apps.

```
<add key="ida:AuthorizationUri" value="https://login.microsoftonline.com" />

<add key="ida:AADInstance" value="https://login.microsoftonline.com" />

<add key="ida:Domain" value="O365_DOMAIN" />

<add key="ida:TenantId" value="TENANT_ID" />
```

```
<add key="ida:UserName" value="USER_NAME" />

<add key="SPOUserName" value="SPO_USER_NAME" />

<add key="ida:PostLogoutRedirectUri" value="HEALTH_CARE_PORTAL_URL" />

<add key="ida:MeetingSubject" value="MEETING_SUBJECT" />

<add key="TrustedApi" value="TRUSTED_API_URL" />

<add key="MobileSiteUri" value="MOBILE_SITE_URL"/>

<add key="DemoUserId" value="O365_USER_ID"/>

<add key="EmailServer" value="smtp.office365.com"/>

<!--This setting will turn ON or OFF the keyvault, it is recommended to use KeyVault in
PRODUCTION deployment-->
<add key="IsKeyVaultEnabled" value="false" />
<!--Encryption key used for encryption the Query parameters, GUID. Optional when
IsKeyValultEnabled = true-->
<add key="EncryptionKey" value="NEW_GUID" />
<!--Encryption Salt a GUID; Optional when IsKeyValultEnabled = true-->
<add key="EncryptionSalt" value="NEW_GUID" />
<!--Password of the SPO User (SpoUserName) , Optional when IsKeyValultEnabled = true-->
<add key="SpoPassword" value="SPO_USER_PASSWORD" />
```

5 Azure Web Apps Deployment (Continuous)

5.1 Update the Config file

- Open the HealthCarePortal.Sln in Visual Studio 2015
- Make sure the project loads successfully

5.1.1 Update Bot Web.Config

- Click on the **VHCBot** project and expand it
- Open Web.Config file
- Update the following values

```
    <!-- update these with your BotId, Microsoft App Id and your Microsoft App Password-->
     <add key="BotId" value="BOT_Handle" />
     <add key="MicrosoftAppId" value="Microsoft_App_Id" />
     <add key="MicrosoftAppPassword" value="Microsoft_App_Secret" />
     <!--URL of the azure web app where Virtual Health is hosted-->
     <add key="ida:HealthCarePortal" value="https://[resourceName].azurewebsites.net" />
<!--This setting will turn ON or OFF the keyvault, it is recommended to use KeyVault in
PRODUCTION deployment-->
     <add key="IsKeyVaultEnabled" value="false" />
     <!--Key vault base URL, Optional when IsKeyValultEnabled = false-->
     <add key="KeyVaultBaseUrl" value="https://[resourceName].vault.azure.net" />
<!--Encryption key used for encryption the Query parameters, GUID. Optional when
IsKeyValultEnabled = true
Use the same Encryption key as Virtual Health Portal-->
     <add key="EncryptionKey" value="GUID" />
<!--Encryption Salt a GUID; Optional when IsKeyValultEnabled = true
         Use the same Encryption Salt as Virtual Health Portal-->
     <add key="EncryptionSalt" value="GUID" />
<!--Password of the SPO User (SpoUserName) , Optional when IsKeyValultEnabled = true-->
     <add key="SpoPassword" value="Password" />
<!--URL of the Virtual Health SharePoint Site-->
     <add key="SharepointSite"
value="https://[tenantName].sharepoint.com/sites/[siteName]" />
<!--User name to access SharePoint resources, recommended to have a service account
instead of user account-->
     <add key="SpoUserName" value="[userName]@[tenantName].onmicrosoft.com" />
<!--Set it to true if you are deploying this Bot to work with the Angular App
(HealthCare.Web)-->
     <add key="IsAngularApp" value="true" />
```

Save the file and close it

5.1.2 Update HealthCare Portal URL

- Click on the **HealthCare.Portal** project and expand it
- Open **Web.Config** file

Update the following values with values captured in above section (user your notepad or xml notepad)

```
<!--Client ID of the application registered in the Azure AD to authenticate virtual
health portal-->
<add key="ida:ClientId" value="CLIENT_ID" />
<!--Client Secret of the application registered in the Azure AD-->
<add key="ida:ClientSecret" value="CLIENT_SECRET" />
<!--O365 Tenant Domain-->
<add key="ida:Domain" value="xxxx.onmicrosoft.com" />
<!--O365 Tenant ID-->
<add key="ida:TenantId" value="GUID_TENANT_ID" />
<!--SharePoint site collection URL-->
<add key="SharepointSite" value="https://xxxx.sharepoint.com/sites/xxxxxx" />
<!--User name to access SharePoint resources-->
<add key="SPOUserName" value="xxxx@xxxxx.onmicrosoft.com" />
<!--Post logout URR-->
<add key="ida:PostLogoutRedirectUri" value="https://xxxxxx.azurewebsites.net/" />
<!--Meeting subject to create UCAP meetings-->
<add key="ida:MeetingSubject" value="Appointment Details" />
<!--Azure webapp URL where virtual health is hosted-->
<add key="ida:HealthCarePortal" value="https://xxxxxx.azurewebsites.net/" />
<!--Key vault base URL, Optional when IsKeyValultEnabled = false-->
<add key="KeyVaultBaseUrl" value="https://xxxx.vault.azure.net" />
<!--Client ID of the application registered in Azure AD to access Key Vault, Optional
when IsKeyValultEnabled = false-->
<add key="ClientId" value="CLIENT_ID_GUID" />
<!--Thumbprint of the certificate to access Key Vault Optional when IsKeyValultEnabled =
false-->
<add key="Thumbprint" value="CERTIFICATE_THUMBPRINT" />
<!--Trusted Application enpoint, this is cloud/ app service https URL e.g.
https://resourcename.cloudapp.net or https://yourtrustedapp.com or
https://trustedapp.yourorgdomain.com or https://resourcename.azurewebsites.net-->
<add key="TrustedApi" value="HTTPS_URL_AZURE_CLOUD_SERVICE_FOR_TRUSTED_APP" />
<!--Application Insight Key-->
<add key="iKey" value="APP_INSIGHTS_ID" />
<!--Bot embed URI-->
<add key="botUrlEmbed" value="https://webchat.botframework.com/embed/xxxxxx?s=xxxxxx" />
<!--Lamna Web API URI, optional-->
<add key="MobileSiteUri" value="" />
<!--Email server to send email-->
<add key="EmailServer" value="smtp.office365.com" />
<!--Admin group to access the configuration page of the Virtual Health-->
<add key="SharepointAdminGroup" value="SHAREPOINT_GROUP_NAME" />
<!--This setting will turn ON or OFF the keyvault, it is recommended to use KeyVault in
PRODUCTION deployment-->
<add key="IsKeyVaultEnabled" value="false" />
<!--Encryption key used for encryption the Query parameters, GUID. Optional when
IsKeyValultEnabled = true-->
```

```
<add key="EncryptionKey" value="NEW_GUID" />
<!--Encryption Salt a GUID; Optional when IsKeyValultEnabled = true-->
<add key="EncryptionSalt" value="NEW_GUID" />
<!--Password of the SPO User (SpoUserName) , Optional when IsKeyValultEnabled = true-->
<add key="SpoPassword" value="SPO_USER_PASSWORD" />
```

- Save the file and close it

- Update the Web.config for the HealthCare.Web Angular application to use the same configuration values. Update **PostLogoutRedirectUri** & ida:**HealthCarePortal** keys to point to HealthCare.Web angular web application.

5.1.3 Update Deployment Tool Config file

Follow the below steps to update the config file of deployment tool (Download from here)

1. Open **DeploymentTool.exe.config** file
2. Update the appsettings values as per environment
   Path of

```
<add key="basePath" value="SHAREPOINT_SITE_URL"/>
<add key="username" value="SPO_USER_NAME"/>
<add key="password" value="SPO_USER_PASSWORD"/>
```

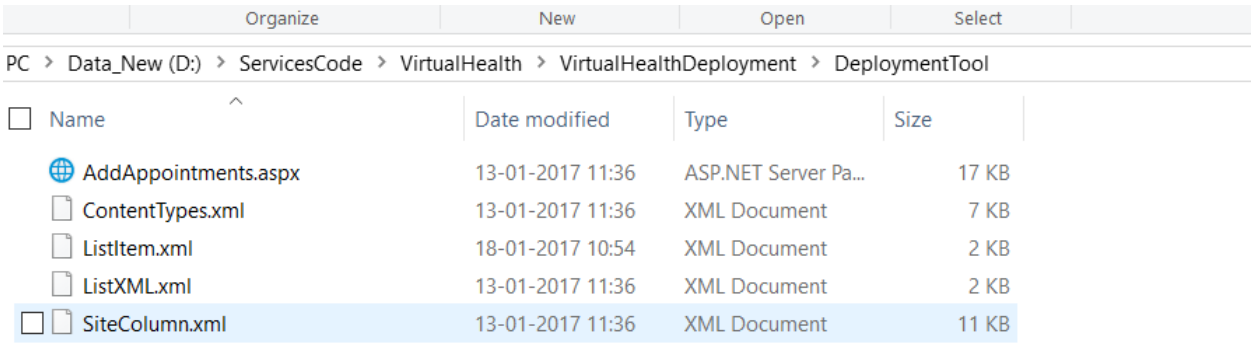
3. Save it

5.2 SharePoint Deployment

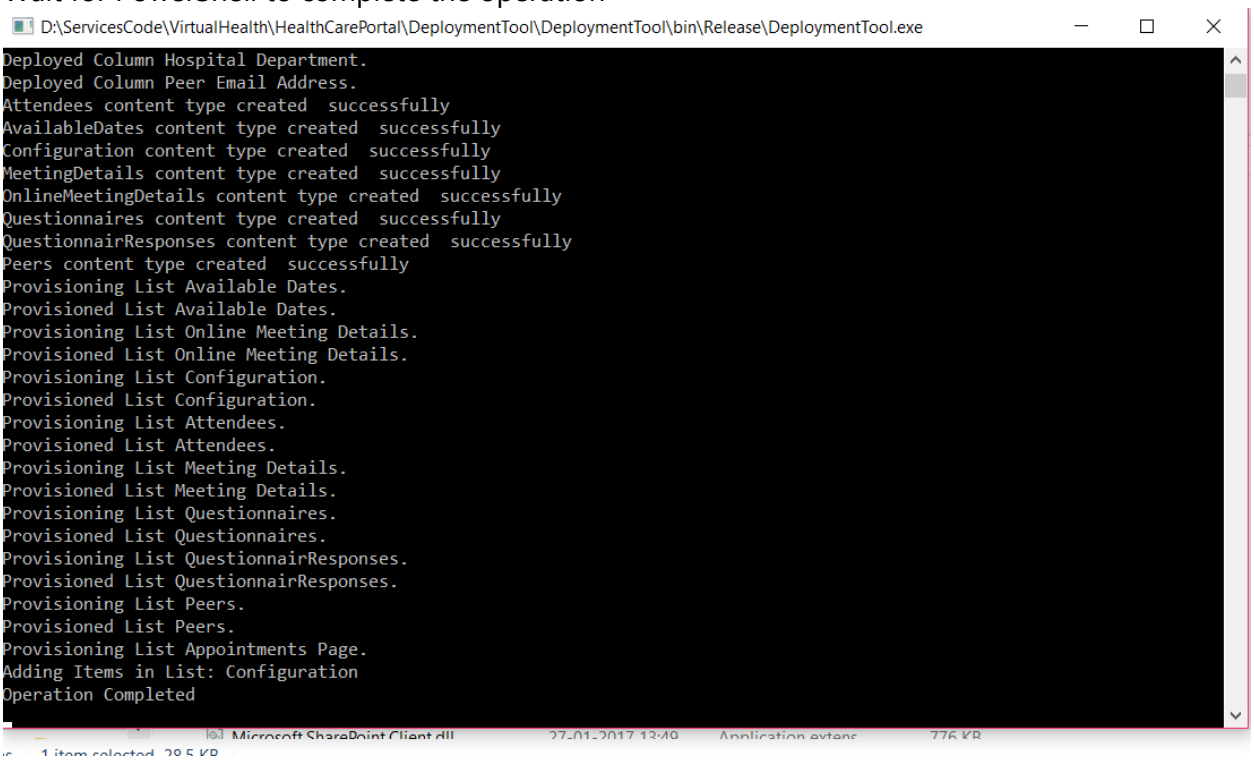
Deploy the SharePoint artefacts using below steps

- Download the deployment tool
- The folder would contain following file

DeploymentTool.exe	30-11-2016 21:11	Application	28 KB
DeploymentTool.exe.config	28-11-2016 10:36	XML Configuration...	1 KB
DeploymentTool.pdb	30-11-2016 21:11	Program Debug D...	60 KB
DeploymentTool.vshost.exe	30-11-2016 21:11	Application	23 KB
DeploymentTool.vshost.exe.config	28-11-2016 10:36	XML Configuration...	1 KB
Microsoft.SharePoint.Client.dll	04-07-2014 04:43	Application extens...	554 KB
Microsoft.SharePoint.Client.Runtime.dll	04-07-2014 04:43	Application extens...	288 KB



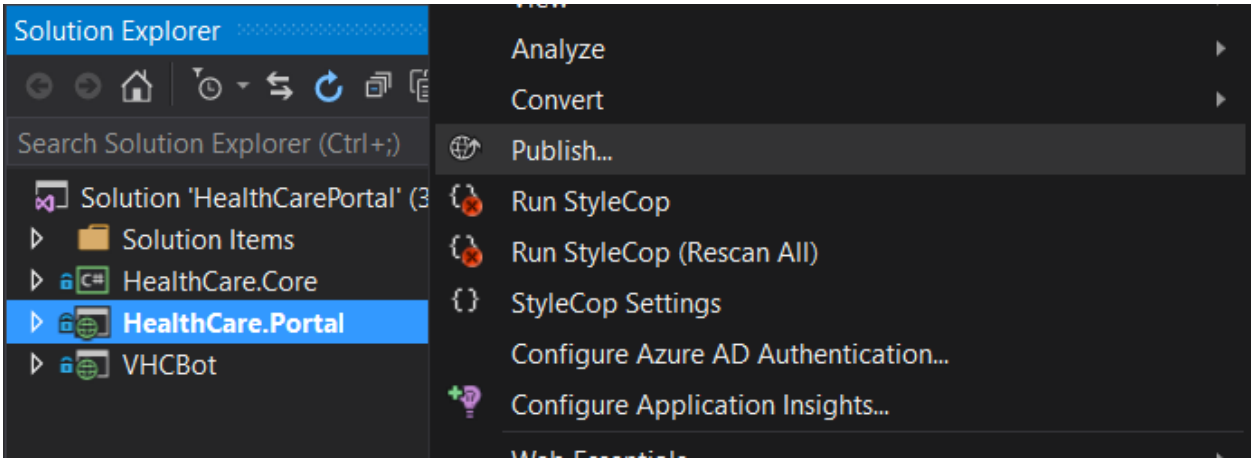
- Double click DeploymentTool.exe
- Select Option 6
- Wait for PowerShell to complete the operation



5.3 Virtual Health Deployment

There are multiple ways to deploy a Web App in Azure. This section talks about Visual Studio Web Deploy publishing.

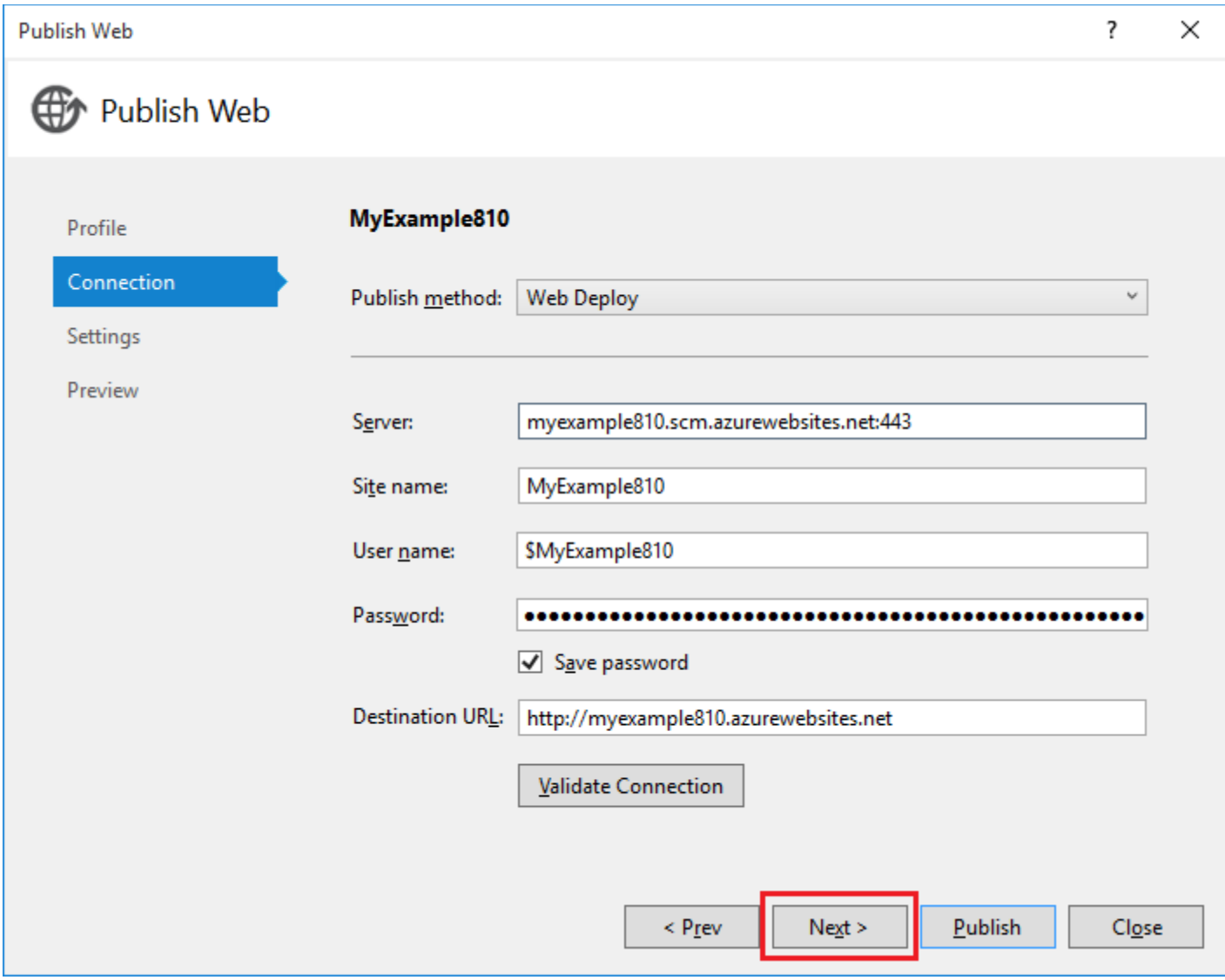
In **Solution Explorer**, right-click the project, and choose **Publish**.



In a few seconds, the **Publish Web** wizard appears. The wizard opens to a *publish profile* that has settings for deploying the web project to the new web app.

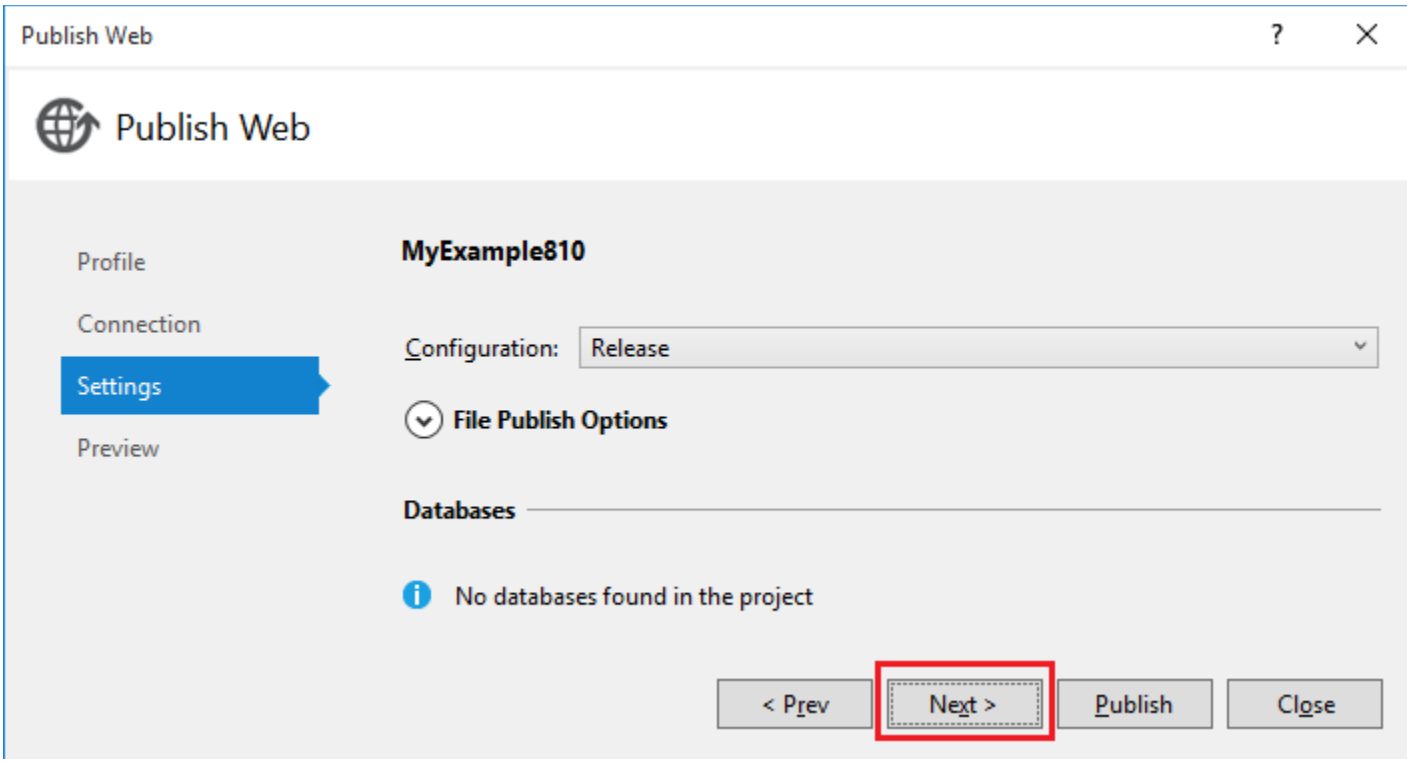
> Tip
>
> The publish profile includes a user name and password for deployment. These credentials have been generated for you, and you don't have to enter them. The password is encrypted in a hidden user-specific file in the `Properties\PublishProfiles` folder.

1. On the **Connection** tab of the **Publish Web** wizard, click **Next**.



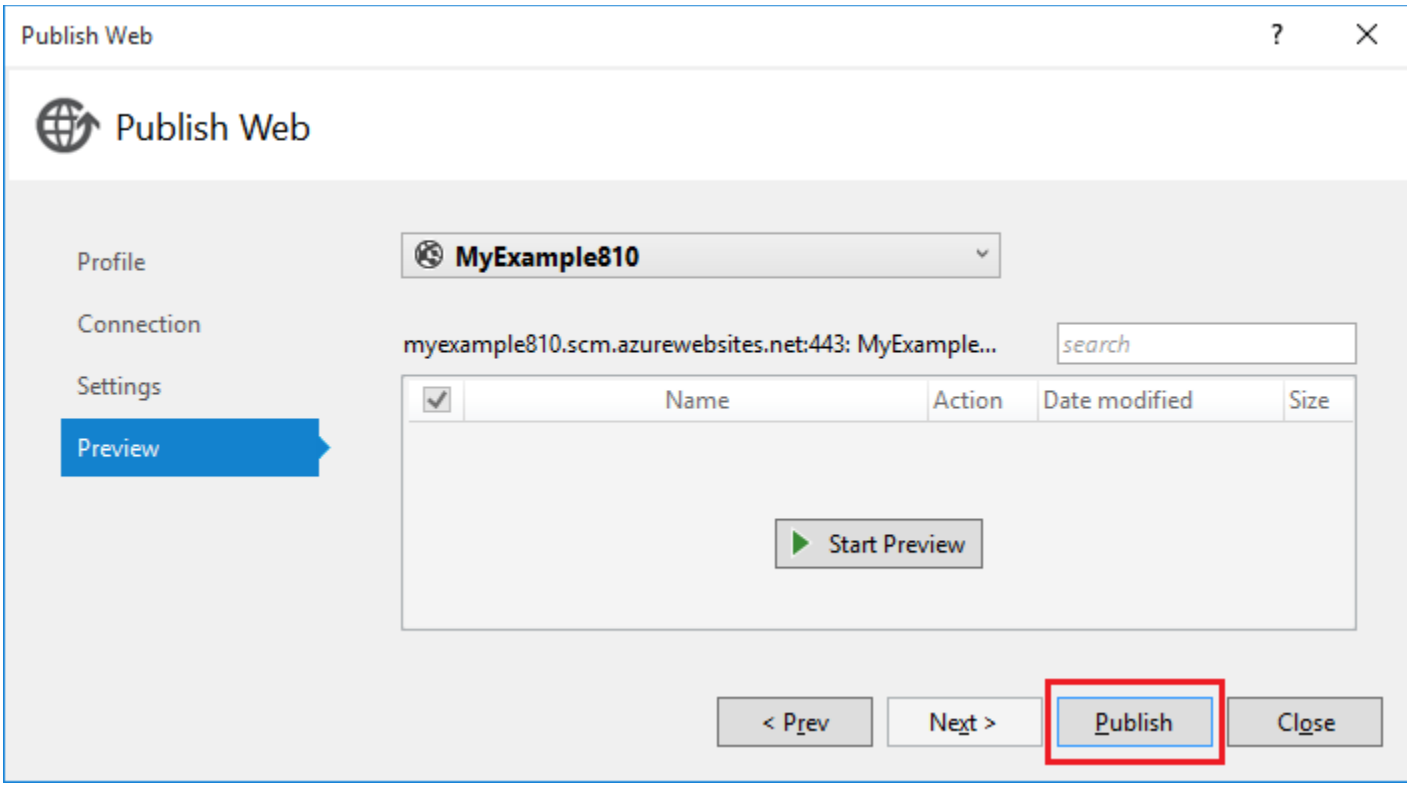
Next is the **Settings** tab. Here you can change the build configuration to deploy a debug build for remote debugging. The tab also offers several File Publish Options.

2. On the **Settings** tab, click **Next**.



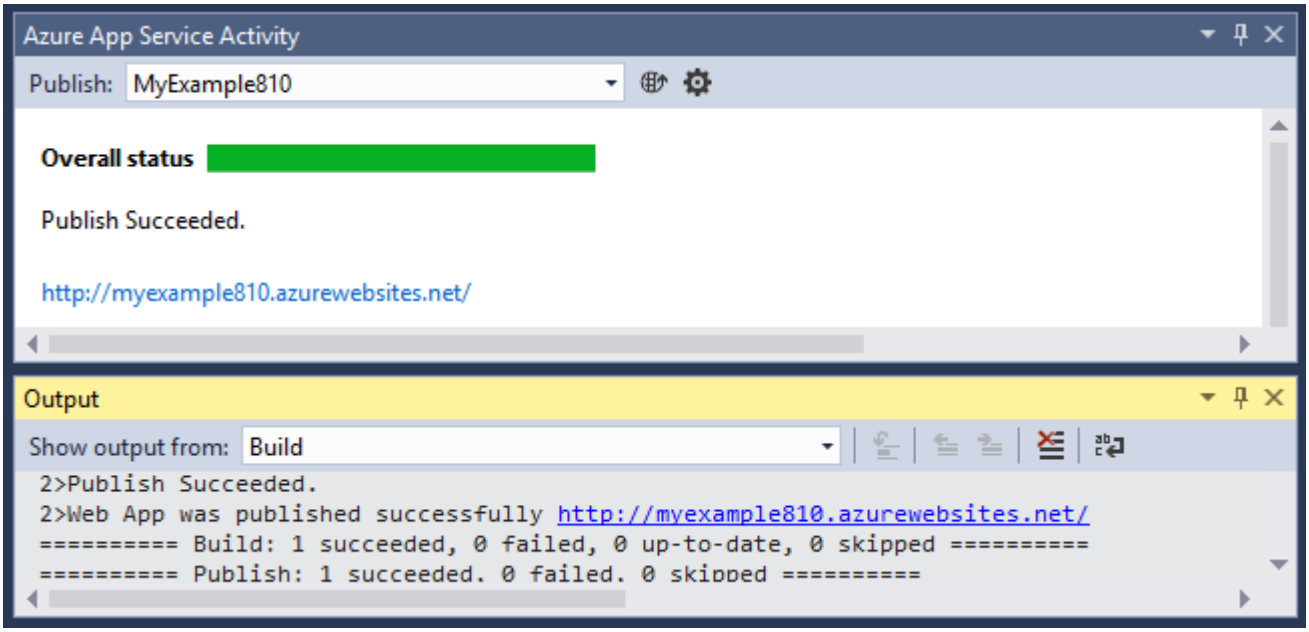
The **Preview** tab is next. Here you have an opportunity to see what files are going to be copied from your project to the API app. When you're deploying a project to an API app that you already deployed to earlier, only changed files are copied. If you want to see a list of what will be copied, you can click the **Start Preview** button.

3. On the **Preview** tab, click **Publish**.



When you click **Publish**, Visual Studio begins the process of copying the files to the Azure server. This may take a minute or two.

The **Output** and **Azure App Service Activity** windows show what deployment actions were taken and report successful completion of the deployment.



Upon successful deployment, the default browser automatically opens to the URL of the deployed web app, and the application that you created is now running in the cloud. The URL in the browser address bar shows that the web app is loaded from the Internet

5.4 Virtual Health Angular App Deployment

Install the node_modules required by the HealthCare.Web Angular app using npm.

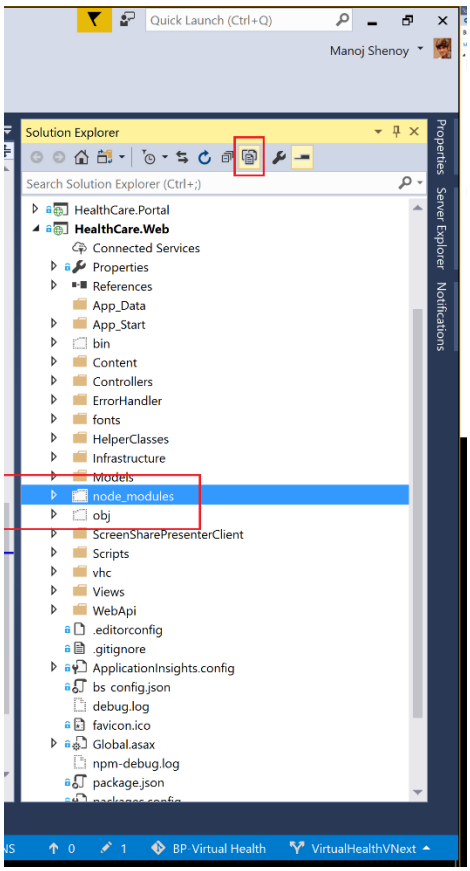
To install node modules, open command window inside the HealthCare.Web directory folder and run the below command

npm install

Once the node_modules are installed, publish HealthCare.Web as detailed in Section 5.3.

To publish the node_modules to web app follow the below steps

i. Go to solution explorer and click on 'Show all files' option
ii. Scroll down to the node_modules folder and publish it by right-click -> publish option



5.5 VHC Bot Web App Deployment

Follow the same step as given in Section 5.3 to deploy VHC Bot Web App.

5.6 Post Deployment Configuration

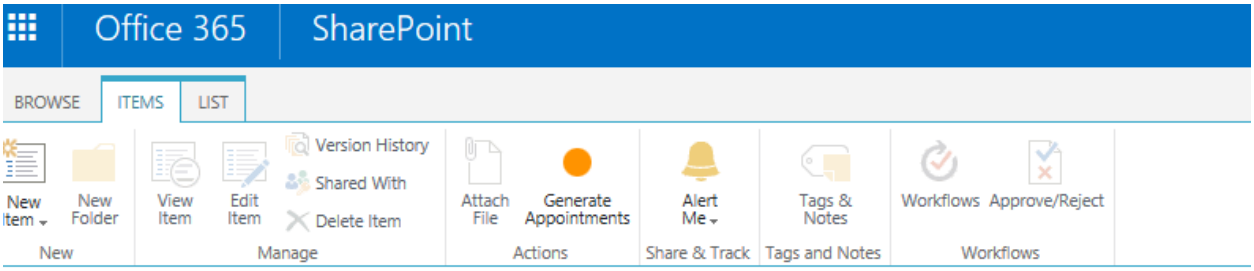
5.6.1 BotUrlEmbed in Web.Config file of HealthCare.Portal

- Go to section 4.4 and Get URL Embed code
- Note down the BotUrlEmbed
- Update the Web.Config of HealthCare.Portal with above value
- Republish the HealthCare.Portal project again using Visual Studio

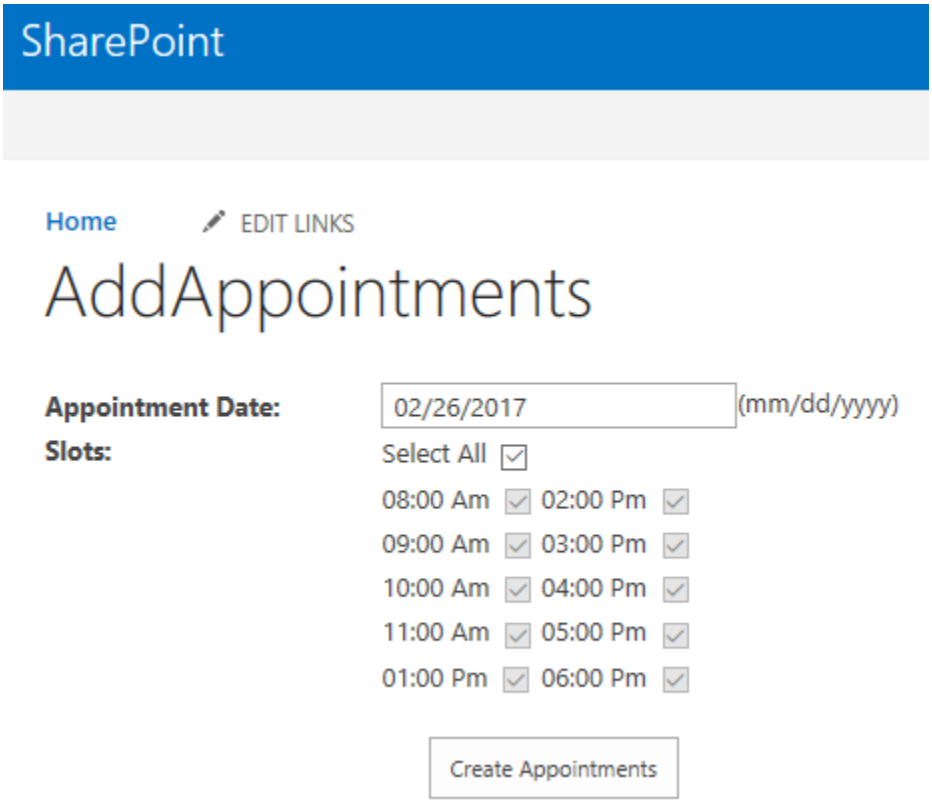
5.6.2 Generate the Meeting Slots

- Open the SharePoint Site Collection created in section 4.1
- Login as site collection administrator
- Go to **Settings → Site Contents**

- Click on "**Available Dates**" list
- On bottom, left corner, there will be a link to change to classic view, click on "**Return to classic SharePoint**"
- Click on **Items** tab and then click on **Generate Appointments**



- Fill in the details as shown below (Date and Check Select All Checkbox)

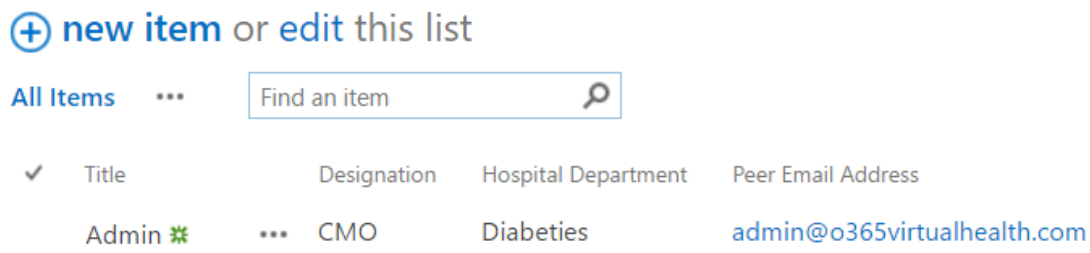


- Click **Create Appointments**
- It will show confirmation message

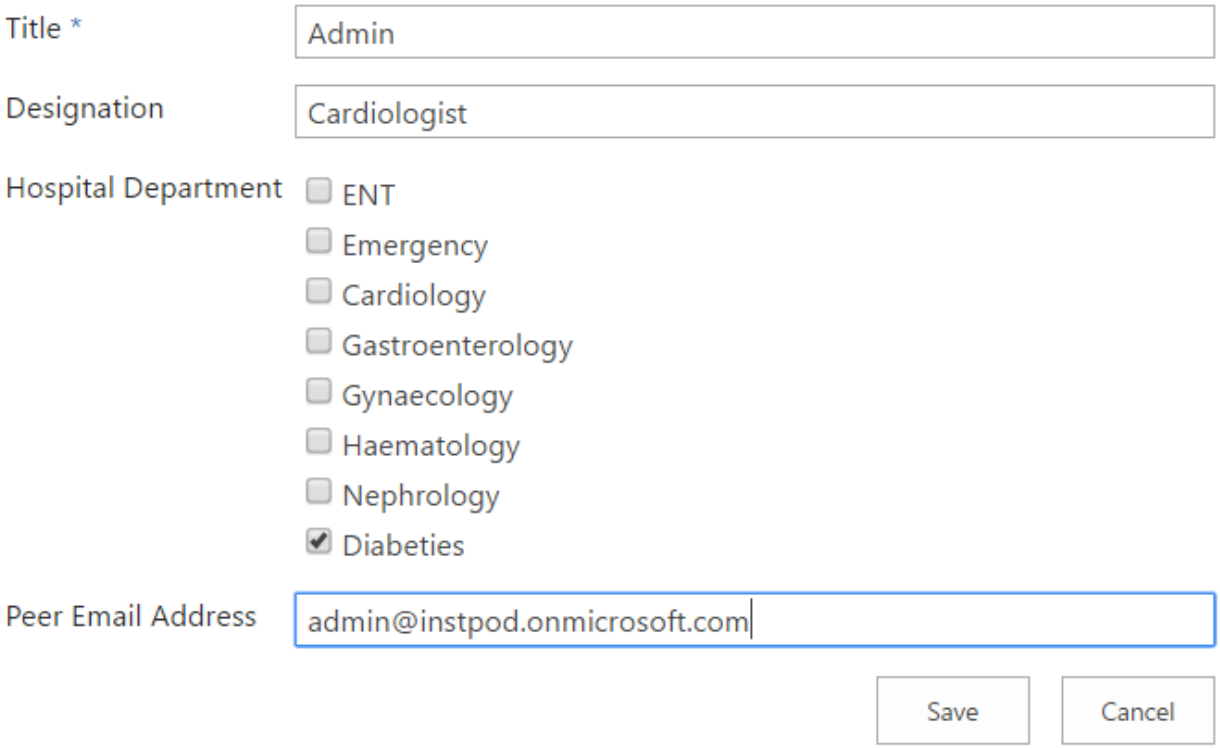
5.6.3 Populate Peers List

- Open the SharePoint Site Collection created in section 4.1
- Login as site collection administrator
- Go to **Settings → Site Contents**
- Click on "**Peers**" list

- On bottom, left corner, there will be a link to change to classic view, click on "**Return to classic SharePoint**"
- Click on "**new item**"



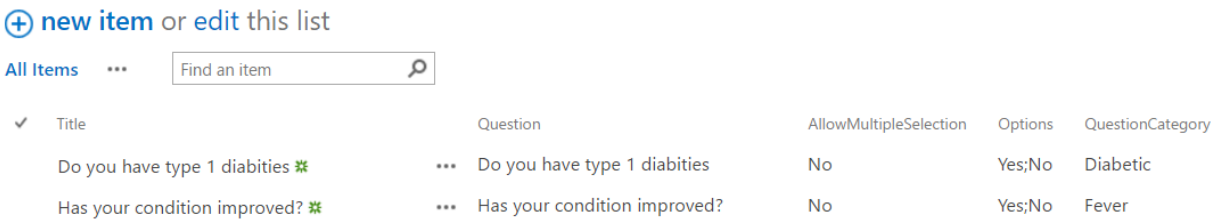
- Add details of admin and other doctors as below and click on "**Save**"



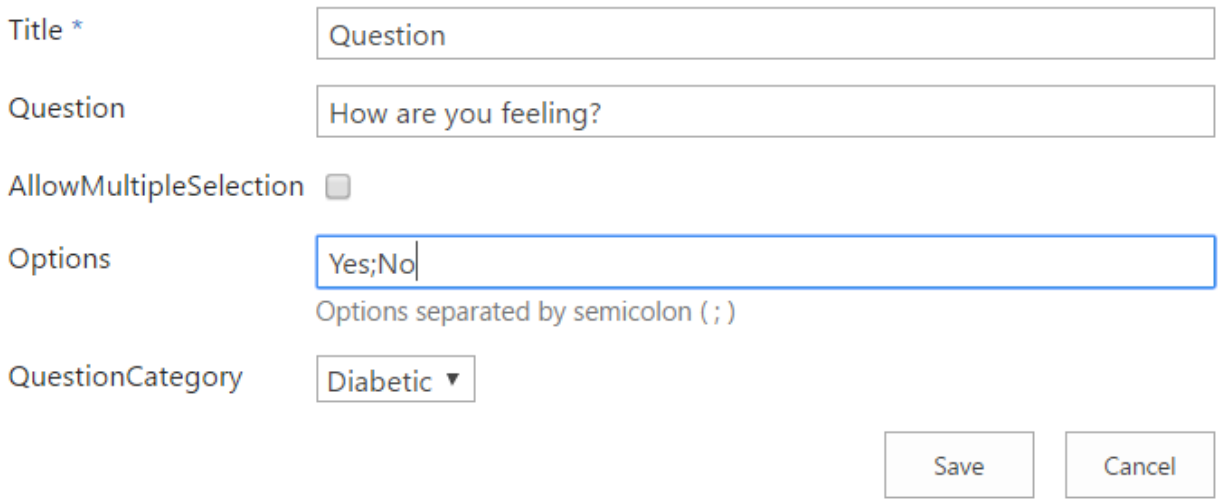
5.6.4 Populate Questionnaire List

- Open the SharePoint Site Collection created in section 4.1
- Login as site collection administrator
- Go to **Settings → Site Contents**

- Click on "**Questionnaires**" list
- On bottom, left corner, there will be a link to change to classic view, click "**Return to classic SharePoint**"
- Click on "**new item**"



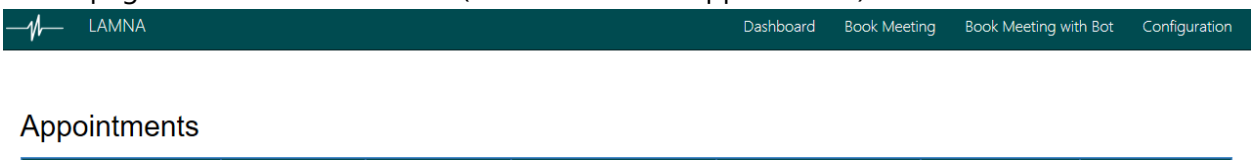
- Add your question to the proper category and click on "**Save**".



6 Post Deployment Validations

Validate Home Page is Opening

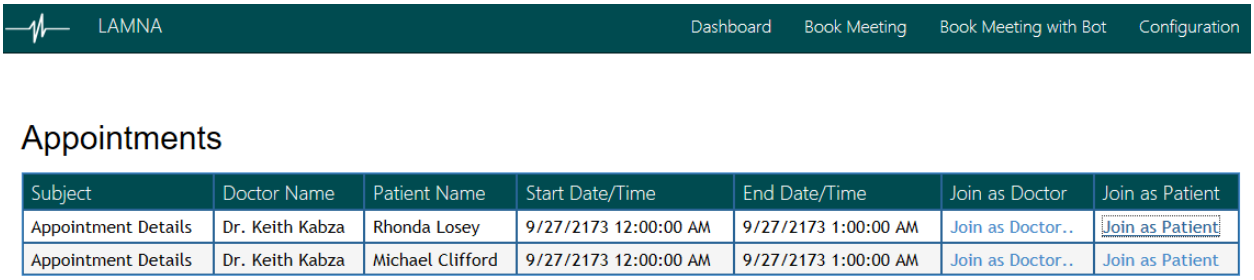
- Open the HealthCare Portal Web App
- Login as Active O365 Tenant user
- Home page should look like below (Since there is no appointment)



Validate Book Meeting

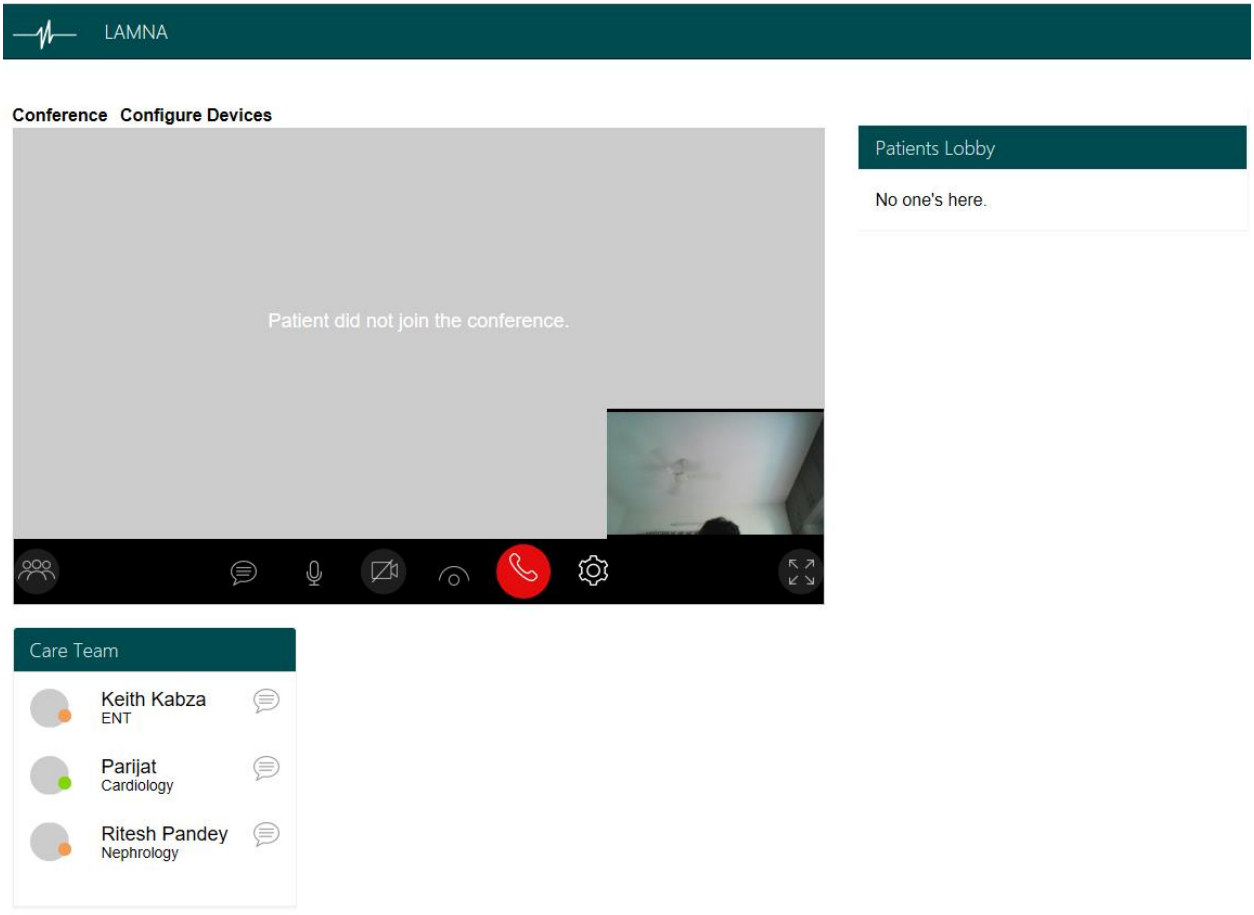
- Click on Book Meeting link on top navigation

- Fill in the details
- Click Submit
- Dashboard should show the meeting in a grid as shown below



Join a Conference as Doctor

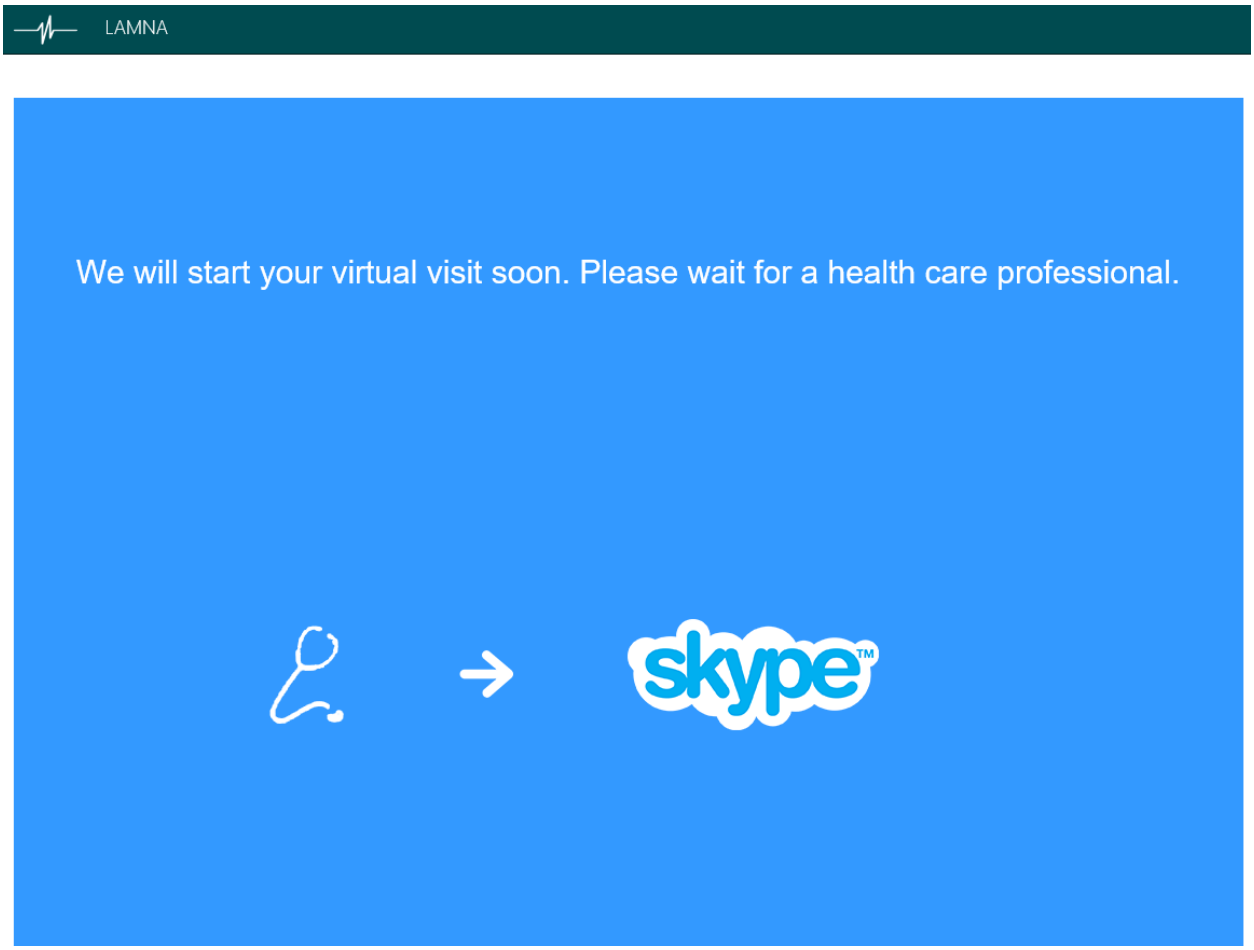
- Home page, Click on the Join as Doctor Link
- It will open the page in a new tab
- Wait for the page to load
- It should look like below



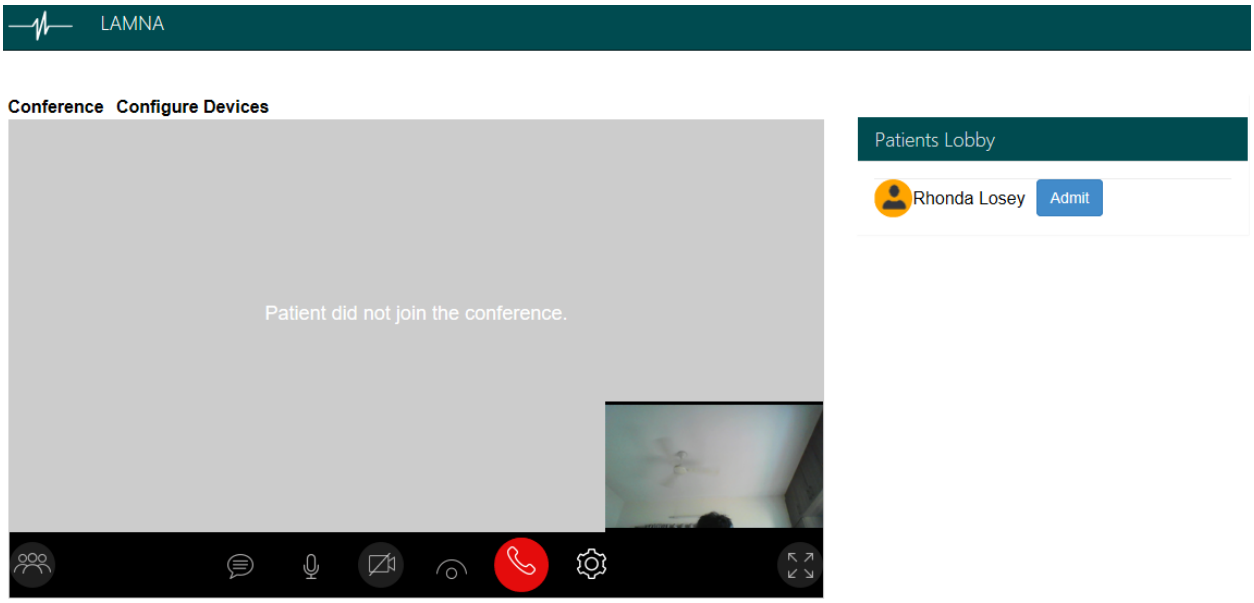
Join Conference as Patient

- Go to dashboard

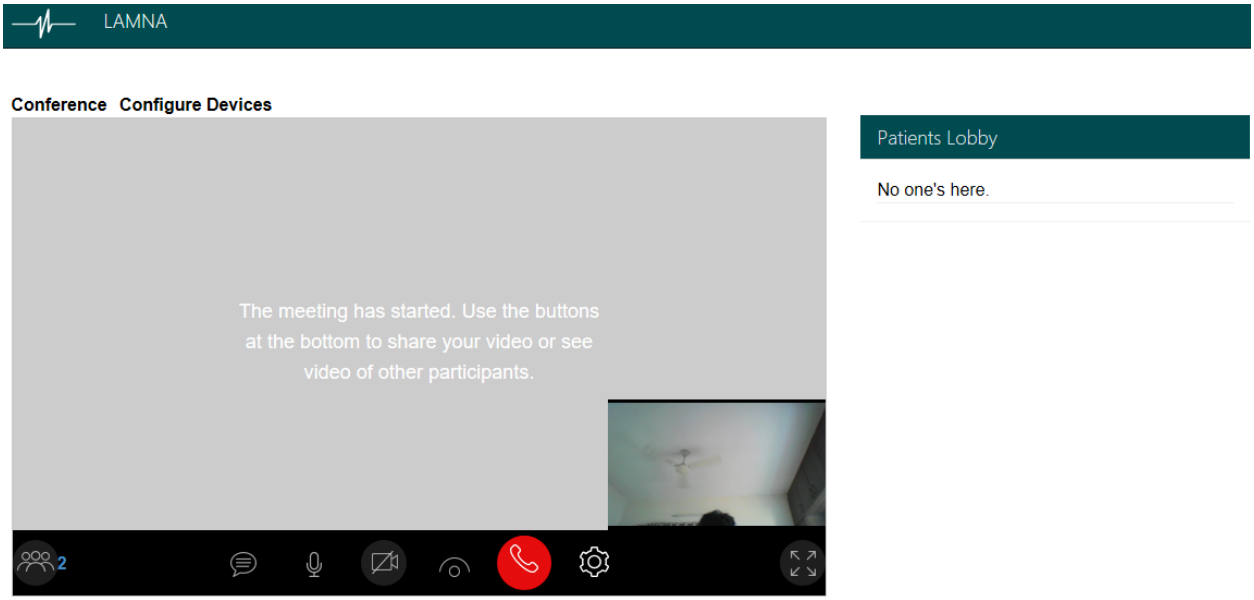
- Join the copy the URL of the patient link from the meeting which Doctor has joined earlier
- Open a new browser window
- Paste the URL
- Wait for the page to load, the patient will wait in lobby



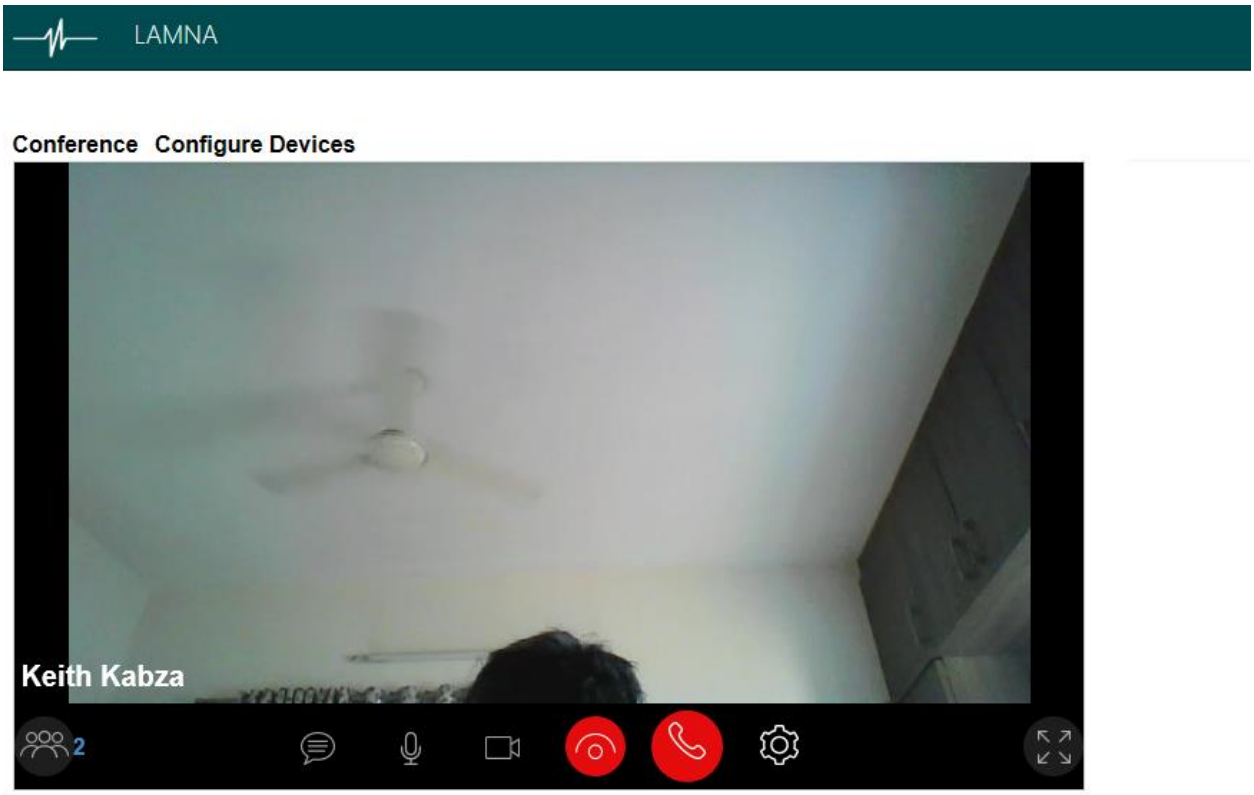
- Go to the doctor window
- Click on Admit



- Meeting starts, and doctor window will look like below

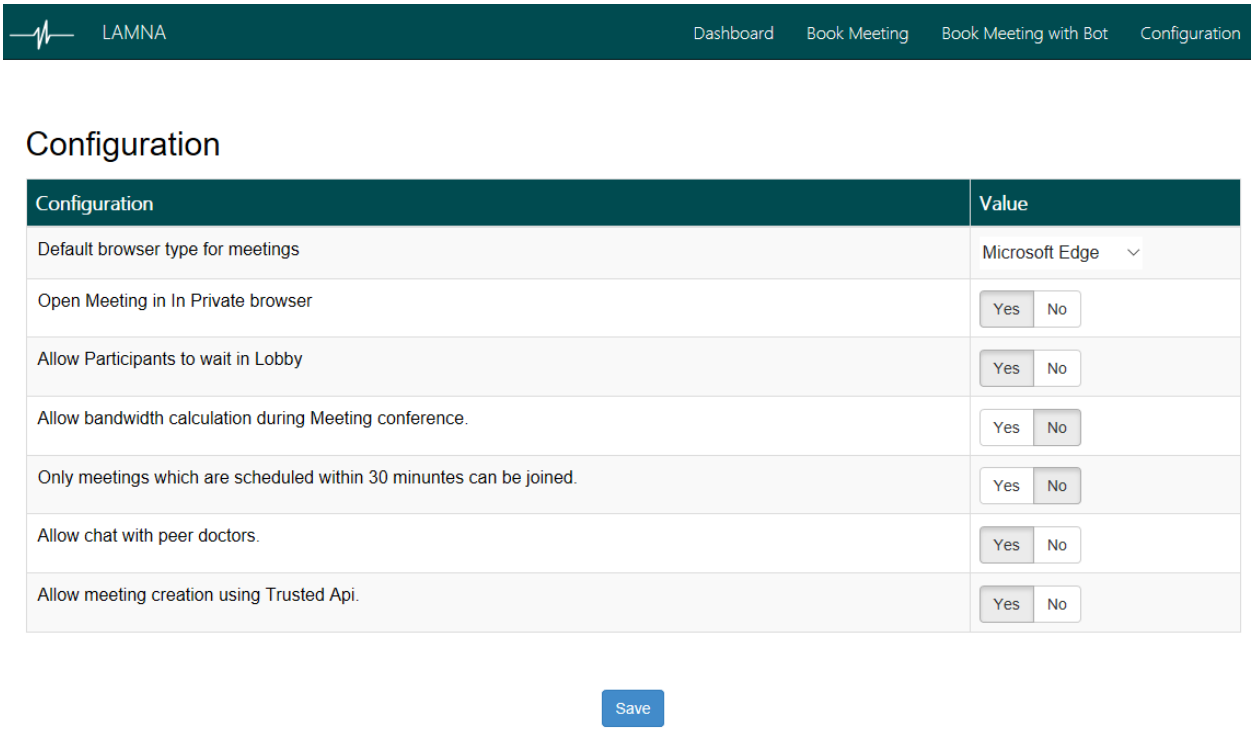


- Go to Patient Window, it will look like below



Verify Configuration Page

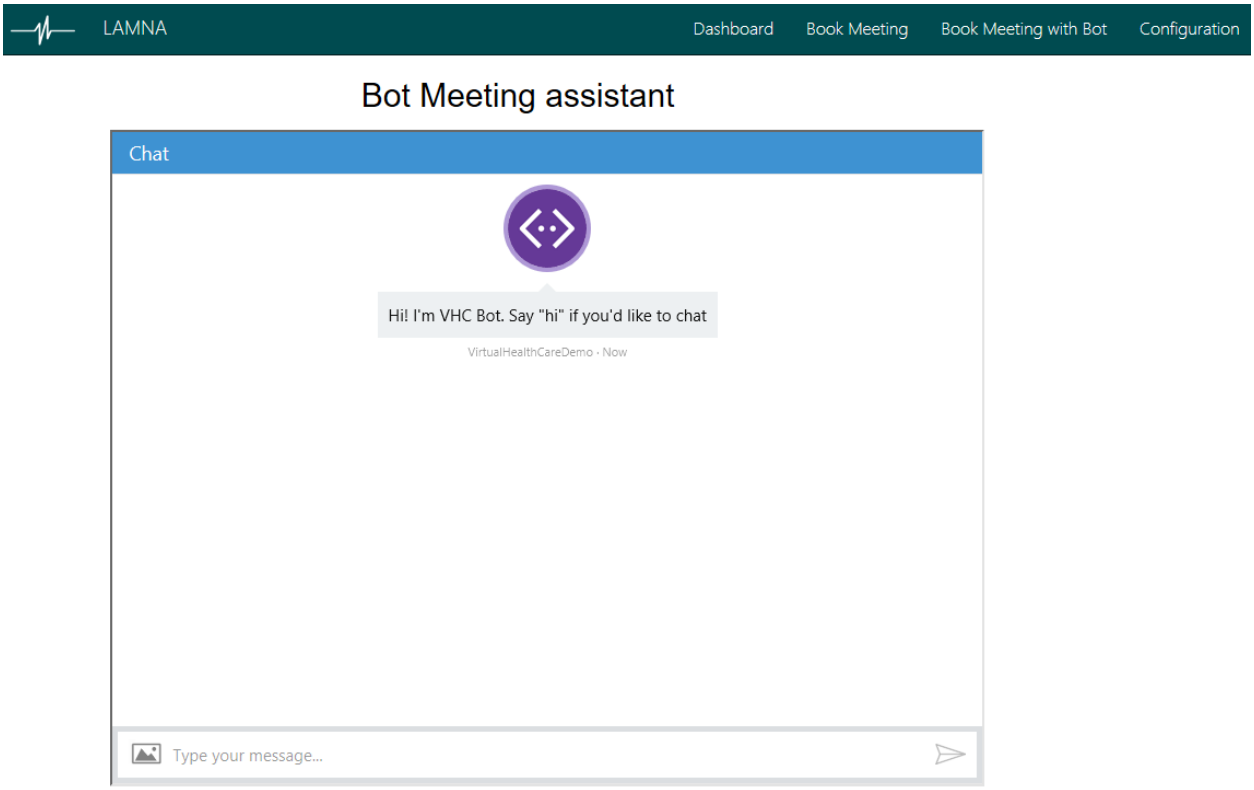
- Click on the Configuration link available in the navigation
- If the user is member of "**VirtualPatientCare Admin**", then page will load, and user will see as below



- Otherwise, user will see an unauthorized message on the page.

Validate Book Meeting with Bot Page

- Click on the "Book meeting with bot" link available in Global Navigation
- It will load the Bot page as shown below

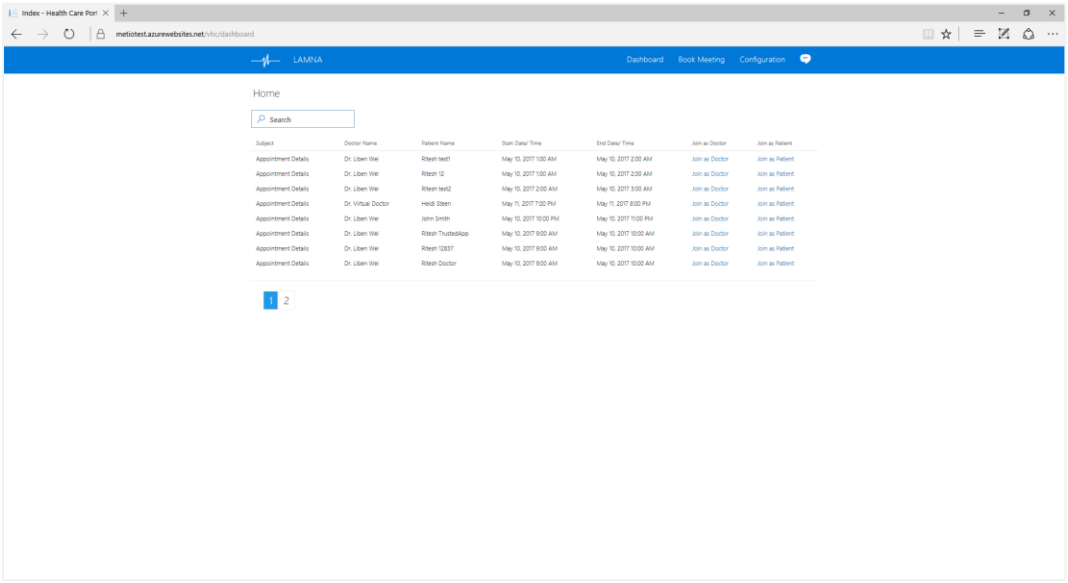


You may try typing below phrases like:
"Book appointment for tomorrow",
"Book appointment on 10/27/2016",
"Cancel my appointment"

☐Below Validation steps are only for Virtual Health Angular Web App

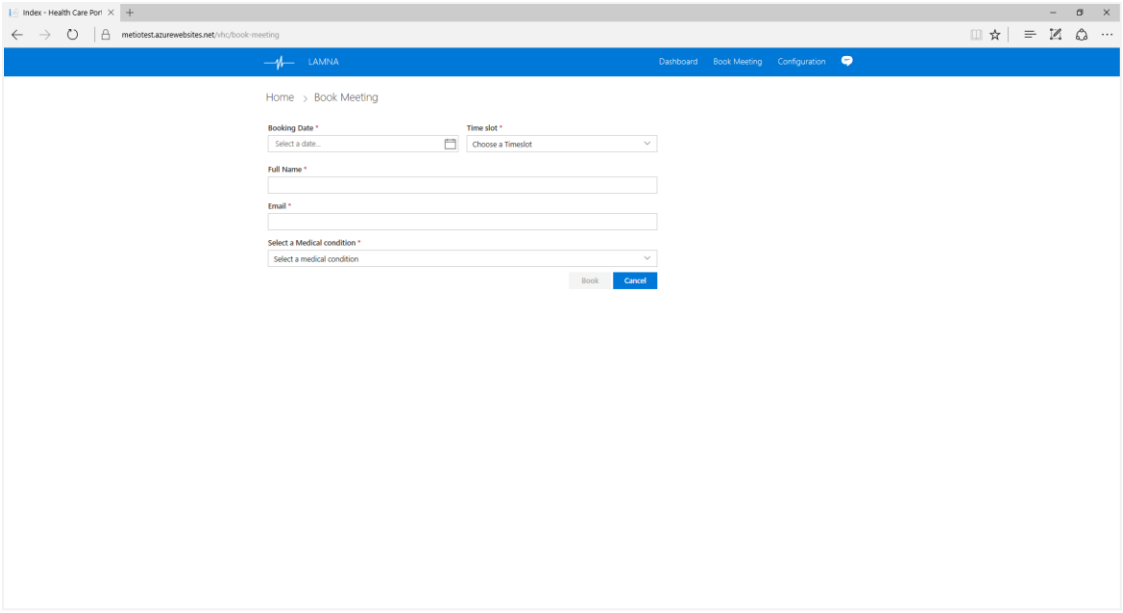
Validate Heath Care Angular App Home Screen

- Login to the Health Care Angular App
- User to be taken to Dashboard screen with list of appointments.



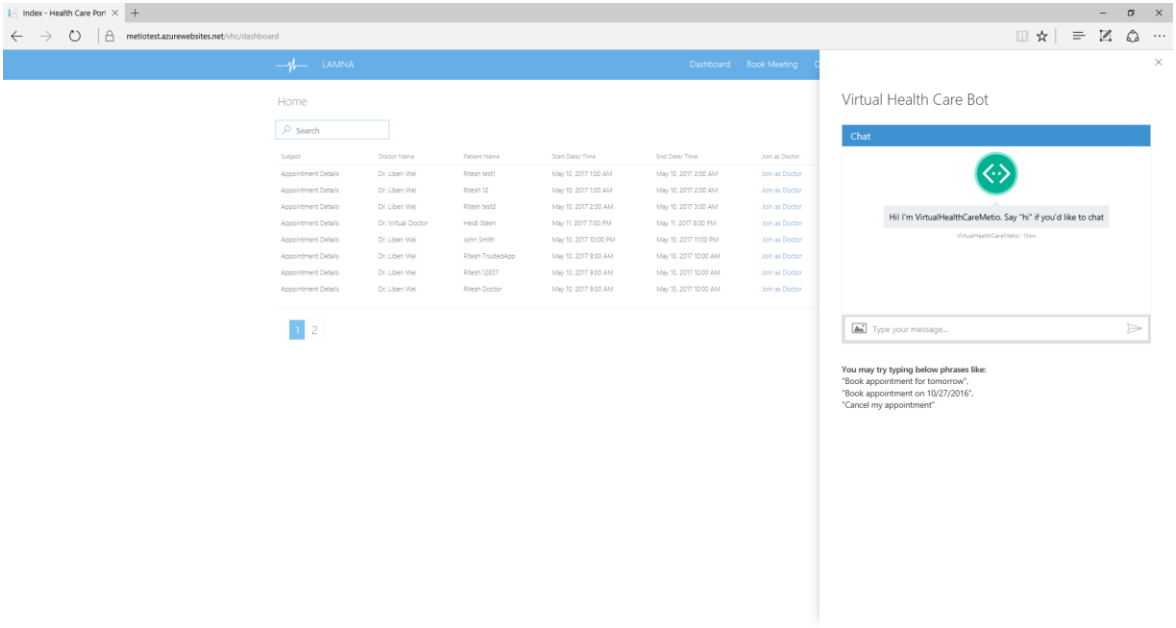
Validate Heath Care Angular App – Book Meeting

- Click on the book meeting screen
- Fill out the book meeting form to book an appointment.
- If booking is successful, user should get a success message via toast notification.



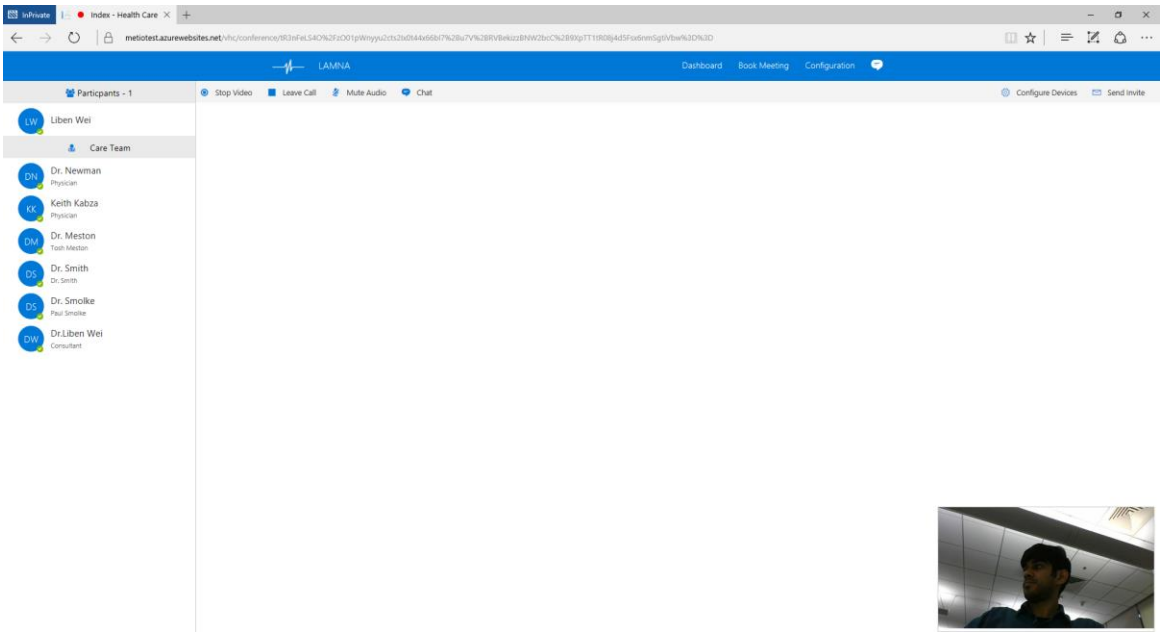
Validate Heath Care Angular App – Book Meeting via Bot

- Click on the bot icon to book meeting via bot
- User should be able to book meeting via bot.



Validate Heath Care Angular App – Video Conference

- From the Dashboard screen, click on the "Join as Doctor" to join a meeting as Doctor.
- Meeting starts with self-video shown on bottom right.


InPrivate
Index - Health Care
metiotest.azurewebsites.net/vhc/conference/tR3nFeL54O%2FzO01pWnyyu2cts2tx0t44x66bI7%2Bu7V%2BRVBekizzBNW2bcC%2B9XpTT1tR0Bj4d5Fox6nmSgtiVbw%3D%3D
LAMNA
Dashboard
Book Meeting
Configuration
Particpants - 1
Stop Video
Leave Call
Mute Audio
Chat
Configure Devices
Send Invite
Liben Wei
Care Team
Dr. Newman
Physician
Keith Kabza
Physician
Dr. Meston
Tooth Meston
Dr. Smith
Dr. Smith
Dr. Smolke
Paul Smolke
Dr.Liben Wei
Consultant