1 Introduction

This document walks you through the steps of deploying Virtual Health Solution and its 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

2 System Requirements

2.1 Office 365 Plan Requirements

Office 365 Enterprise E3 and above are recommended.

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

Note: For developers who build solutions that extend Office and the Office 365 experience across desktop, web, and mobile platforms, you may wish to join Office 365 developer program: https://dev.office.com/devprogram. Benefits include a one-year, FREE Office 365 developer subscription and many more.

2.2 Browser Requirement

The preferred browser is Microsoft Edge on Windows 10. Otherwise, the Skype for Business plugin may not work correctly or may need additional configuration.

For all testing, it's recommended to use a Windows client with the Microsoft Edge browser while choosing In Private Session.

2.3 Azure Subscription

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

- Website/Web App
- Application Insight
- Key Vault
- Azure Table

Note: For basic test deployments, the 30-day FREE trial is sufficient. If you need additional time after the trial ends, upgrade your trial subscription to Pay-As-You-Go.

2.3.1 Minimum Azure Web App Configuration

The Virtual Health web apps need at least a Standard configuration to cater to the needs of a pilot. However, the Standard tier can be scaled out based on the need of your application.

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

2.3.2 Software Requirements

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

3 Prerequisites

3.1 Office 365 and Azure

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 Office 365 developer program: https://dev.office.com/devprogram offers a one-year, FREE Office 365 developer subscription
Domain(Optional)	Domain for Office 365. This is *optional* for the Virtual Health deployment.
Site Collection	Provision a site collection for Virtual Health, preferably a publishing site.

Azure	Details
Azure Subscription	Azure subscriptions will host the 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 certification authority (CA)-issued SSL certificates for the domain. For example, CA-issued certificates are required for Trusted Application Endpoint configuration and deployment. SSL certificates are *optional* when: • You are *not* using Key Vault and *not* deploying the trusted application endpoint as a cloud service.
Active Directory Integration	Set up and synchronize existing Organization Active Directory on Office 365 portal. http://technet.microsoft.com/en-us/library/hh967642
Domain (Optional)	Domain for an 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 (Optional)	Provision a Key Vault for the Virtual Office Solution. This is *optional* if you are not using Key Vault.
User Account	The user has access to provision and configure services in the Azure PaaS and is the site collection administrator. • Azure Subscription admin or similar role • Office 365 Site Collection administrator
Trusted Application Endpoint	This application MUST be deployed as an Azure cloud/Azure app service before the deployment of Virtual Health Solution. Refer to link Trusted Application API.

3.1.1 Trusted Application Endpoint

☐ Before you proceed with the Virtual Health Solution deployment, you MUST deploy the Trusted Application Endpoint.

Trusted App Sample Code Download Link	Usages

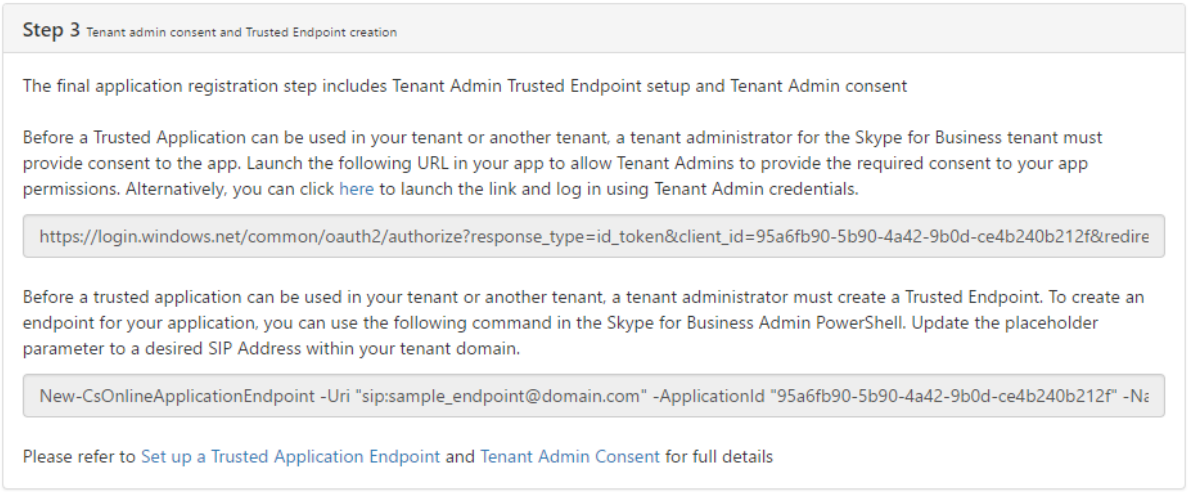
https://github.com/OfficeDev/skype-docs/tree/master/Skype/Trusted-Application-API/samples/AnonMeetingJoinSamples	Download the AnonMeetingJoinSamples and deploy the sample as an Azure cloud service or Azure App Service.
	
	Note down the https endpoint of the Cloud Service e.g. https://yourclouservice.com
	
	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" />`

Note:

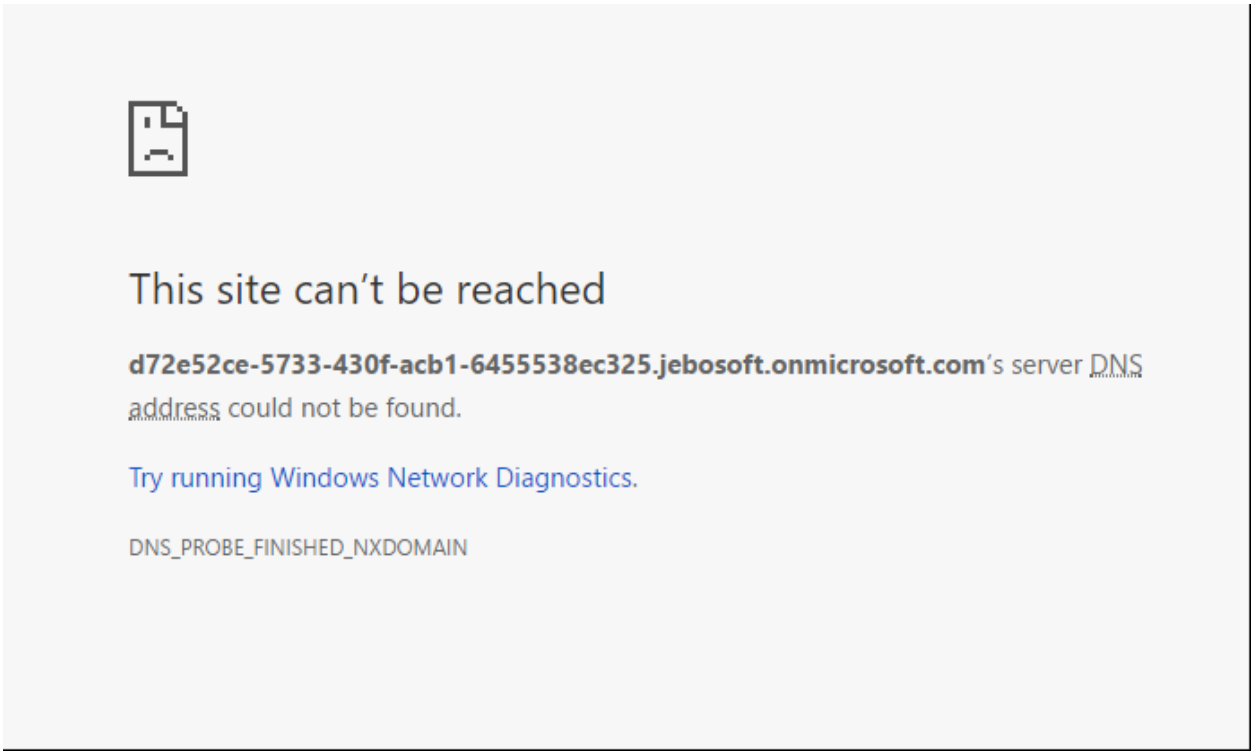
1. AnonMeetingJoinSamples > Getting started > Prerequisite > step 3

https://github.com/OfficeDev/skype-docs/tree/master/Skype/Trusted-Application-API/samples/AnonMeetingJoinSamples#prerequisites

As you go through the steps for the quick registration tool and reach the second to last consent URL on Skype for Business Online Application Registration Portal, consent will be requested.



Once consent is approved, you will see the callback go here. This is the expected behavior.



2. AnonMeetingJoinSamples > Getting started > Prerequisite > step 6

Step 6 is not necessary for deploying the Virtual Health Solution.

3.2 Certificates Required for Deployment

The required certificates for deployment are given in the table below.

Certificate Type	Application	Purpose
Self-signed or **CA issued*** ***Required if Trusted Application is going to use certificate-based authentication (OAuth flow), else client-and-secret-based OAuth flow works fine.**	Trusted Application Endpoints.	This certificate will be used to set up the OAuth with an Azure Active Directory (Azure AD) application.

CA issued* ***To access the cloud service that hosts your apps over https, you need a CA-issued certificate.** **A certificate is *not* required if Trusted Application Endpoint is deployed as an Azure App Service.**	Trusted Application – a cloud service.	This certificate will be required to configure the https endpoint for the cloud service.
Self-signed or **CA Issues*** ***Optional if Key Vault is *not* used for Virtual Health.**	Key Vault Application.	This certificate will be used to set up the Key Vault application access. This certificate will be used in section 4.5.

4 First-Time Provisioning and Configuration

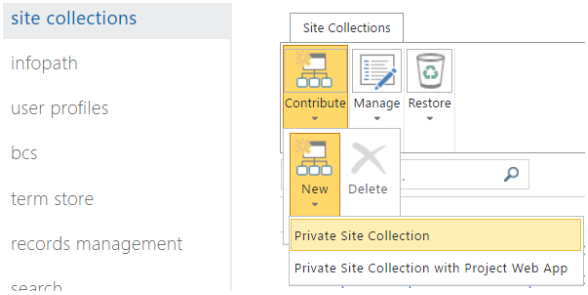
This section provides the steps on how to provision and configure any new environment for the first time. This step is not required once the new environment is set up.

> 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.

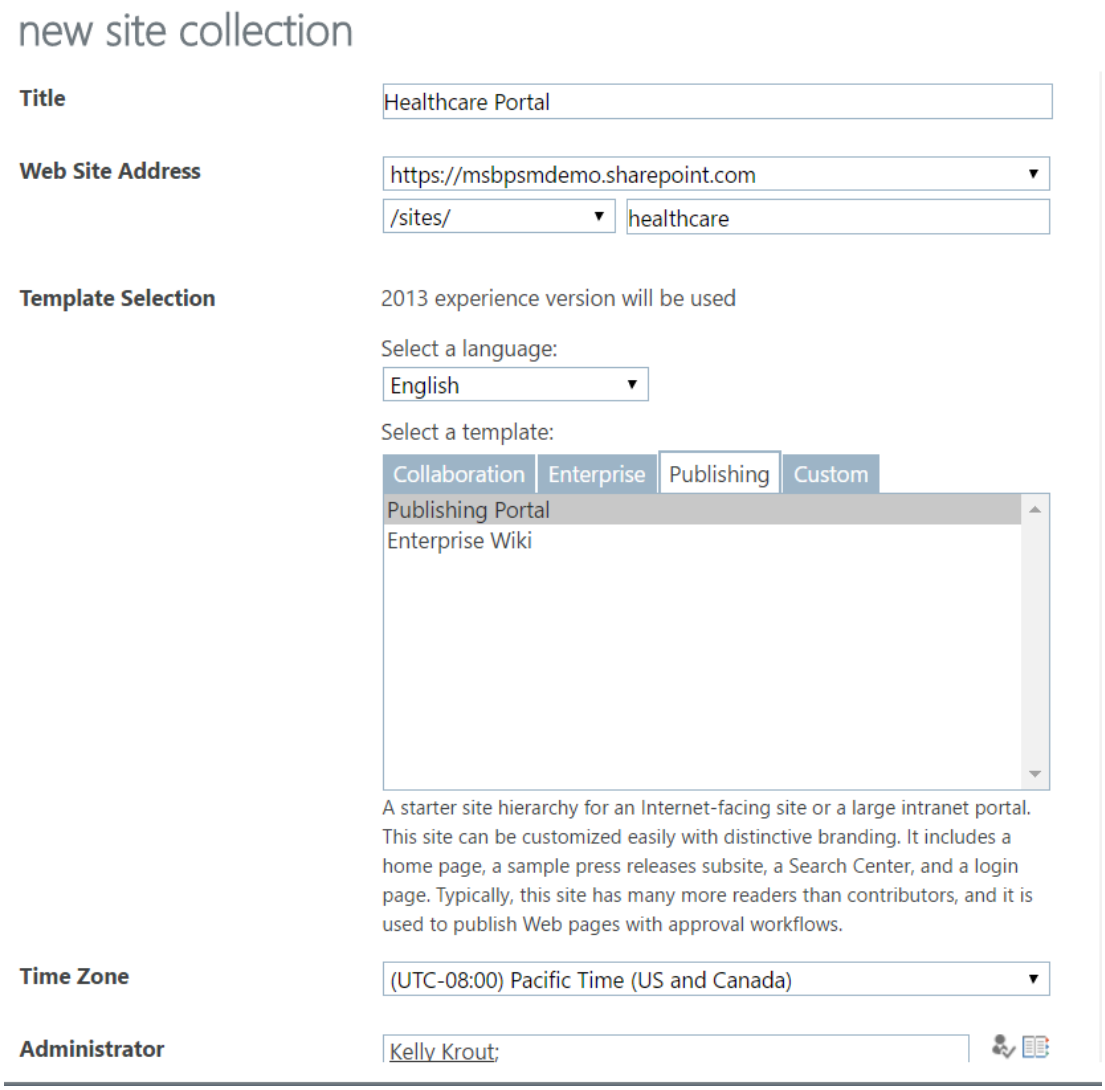
4.1 SharePoint Site Collection Provisioning

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

1. Sign in to the Office 365 admin center with your SharePoint Online admin user name and password.
2. Go to **Admin center**> **SharePoint**.
3. Click on **site collections**.
4. Click on **Contribute** → **New** → **Private Site Collection**



5. Click on **Private Site Collection.**
6. Fill in the details as shown below.



7. Click **Ok**.
8. Note down the site collection's full URL (e.g. https://virtualhealthtest123.sharepoint.com/sites/healthcare) in the format below.

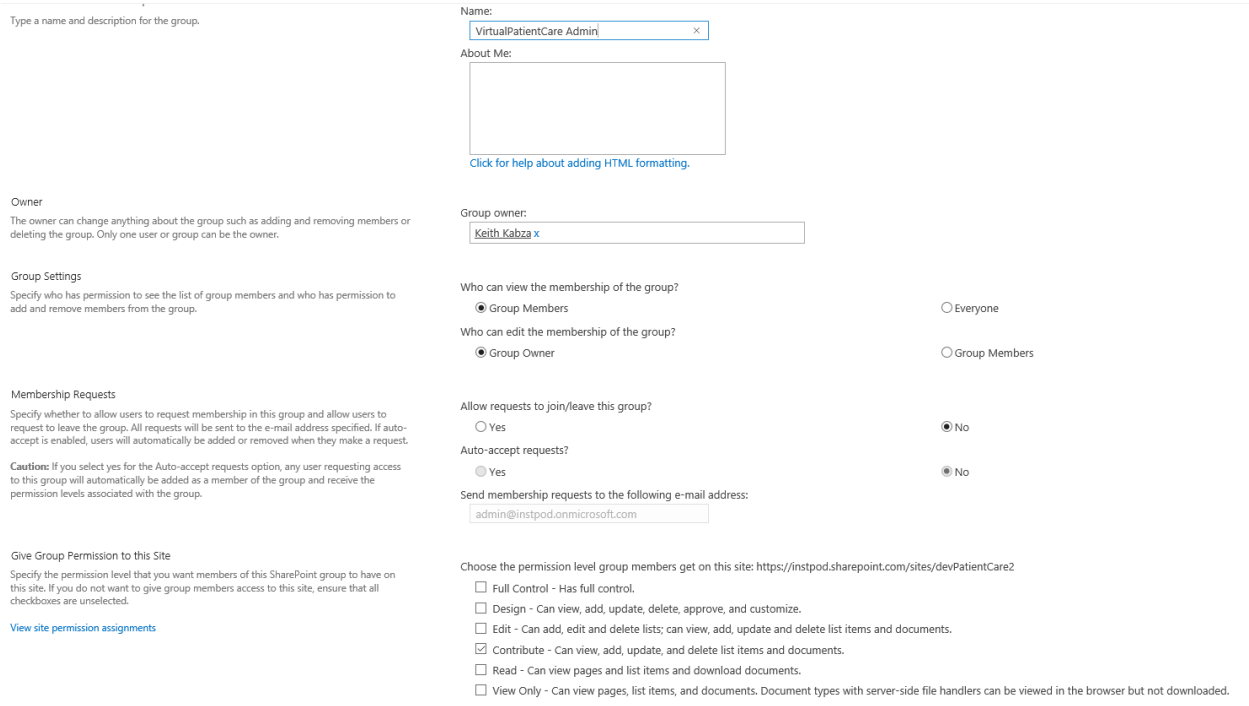
```
<add key="SharepointSite" value="SITE_COLLECTION_URL" />
```

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.

1. Open the site collection.
2. Go to **site settings → People and Groups.**
3. Click on **More.**
4. Click on **New → New Group.**

5. Fill in the details such as Name as "`VirtualPatientCare Admin`" and select **group permission** as **contribute**.



6. Click **Create**.
7. Note down the SharePoint Group Name in the format below.

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

4.3 Azure Web Apps Provisioning

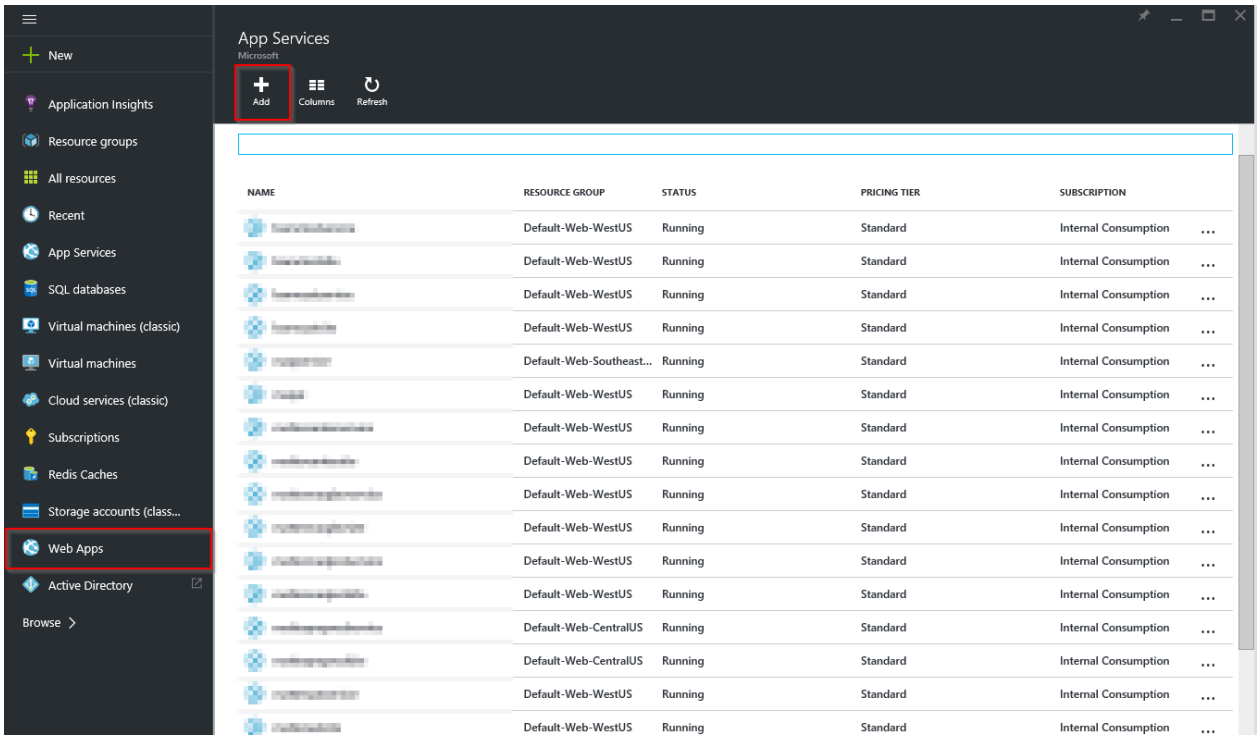
The Virtual Health Solution has the following web apps:

- HealthCare.Portal
- VHCBot

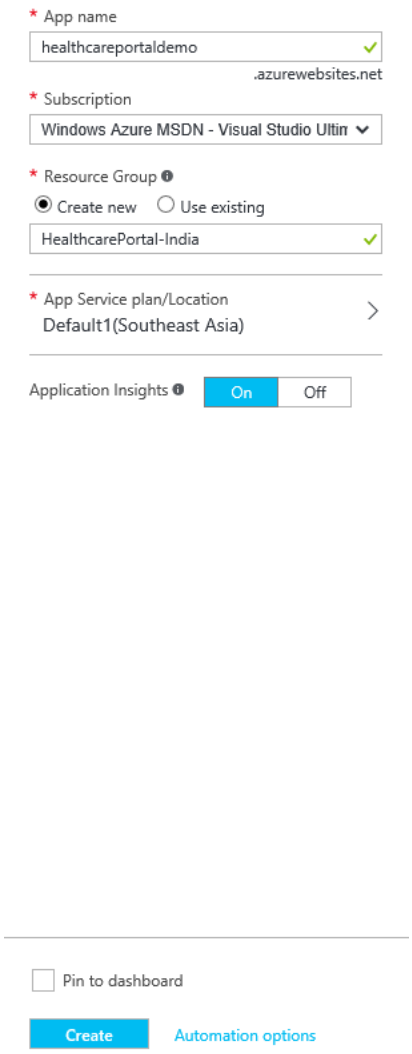
The web apps above can be provisioned using the steps below. As an example, here is a walkthrough of the HealthCare.Portal web app provisioning.

4.3.1 Provision an Azure website

1. Login to the Azure management portal.
2. Select **App Services**, click on the **Add** button, and select **Web App**.



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



4. While creating the website, if you mark the option "Applications Insights" as **On**, Application Insight will be provisioned. Note down the Application Insight key.

5. Click on the **Create** button at the bottom of the panel.

   Note: Select the App Service Location closest to your users. This will help reducing network latency and potentially provide a better experience for users. Southeast Asia is the App Service Location used in the above example. This location will be used to automatically create other services in the same location, so that all required objects are co-located.

4.3.2 Configure Application Insights

In section 4.3.1 step 4, if you marked the option "Applications Insights" as **On** while creating the website, Application Insight will be provisioned. Note down the Application Insight key.

4.3.3 Website – Scaling

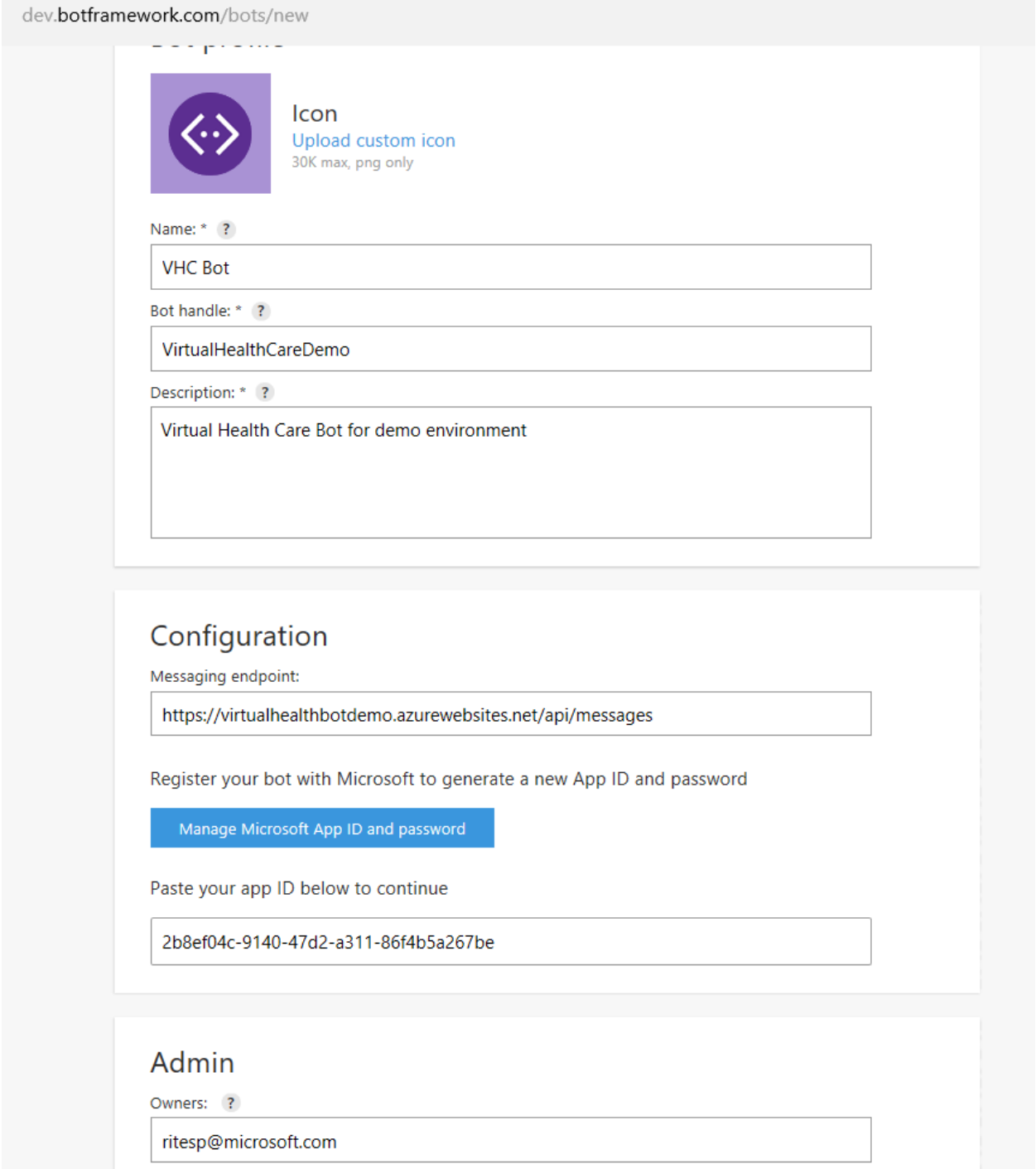
If you want to change the scaling of the website, refer to the prerequisite section. Note down the following values from this section.

```
<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

4.4.1 Register a Bot

1. Go to https://dev.botframework.com
2. Click on **Register a bot**.
3. 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



4. Click on **Create Microsoft App ID and Password**.
   It will generate an App Id and a password. Make a note of them, and update the same items in the Web.config file of Bot Project.
5. Click on **Register** to register the bot.
6. Note down the configurations values in the format below.

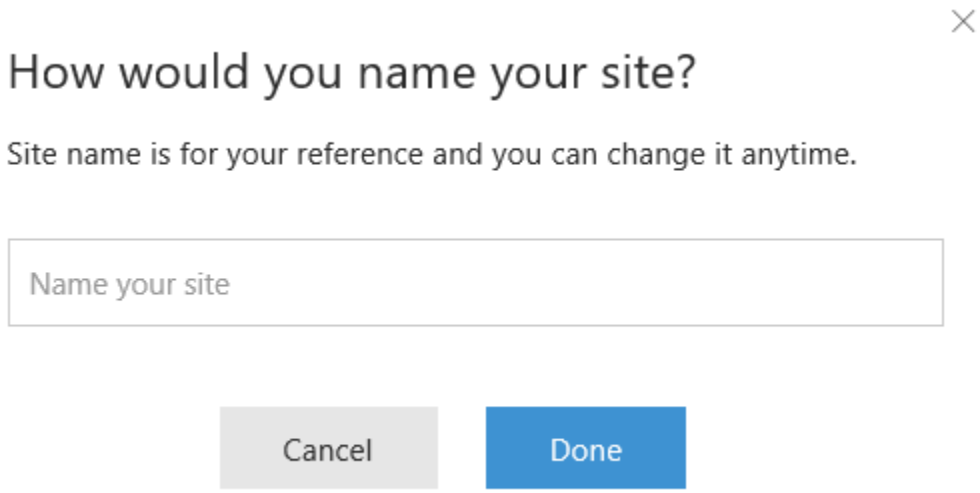
```
<add key="BotId" value="BOT_ID" />

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

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

4.4.2 Get the Bot Embed Code

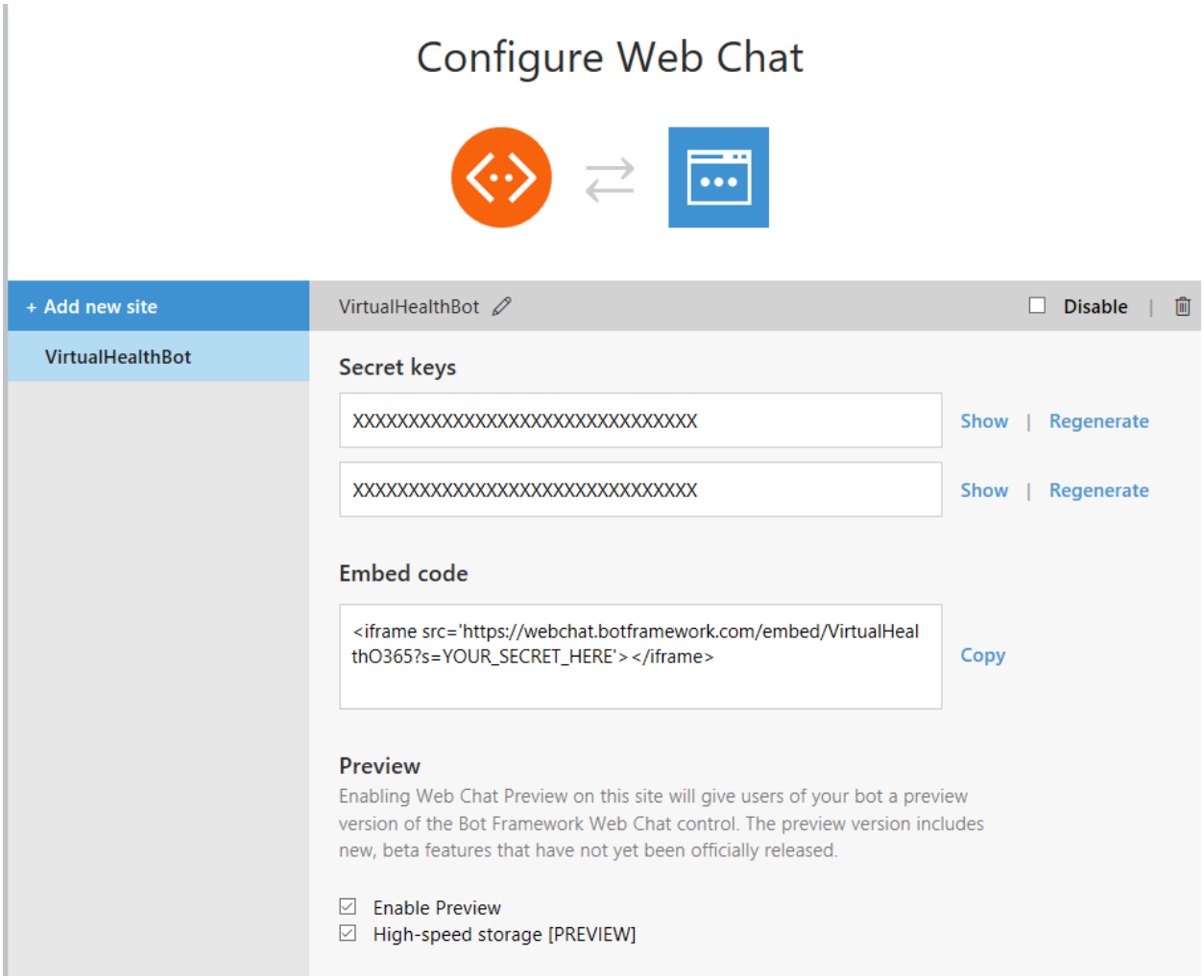
Follow the steps below after completing section 5.4.

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

5. Click on **Add New Site.**
6. Type the name, for example,"VirtualHealthBot".



7. Click **Done**, and you will be redirected to the Configure Web Chat page.



8. Copy the embed URL: https://webchat.botframework.com/embed/VirtualHealthO365?s=YOUR_SECRET_HERE
9. Click on **Show** to reveal the secret, copy it and replace YOUR_SECRET_HERE with the secret.
10. Copy the embed URL. It's required to update the embed URL in the Web.config file of the HealthCare.Portal web project. Note down the configuration value in the format below.

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

4.5 Key Vault Provisioning and Configuration

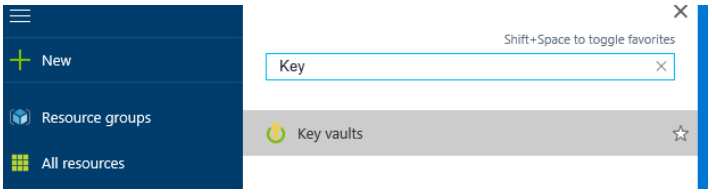
Execute this step if you are going to use Key Vault. If you are using Key Vault, make sure the Web.config of Virtual Health Web App has `<add key="IsKeyVaultEnabled" value="true" />`, otherwise, keep this value equal to false.

For Key Vault provisioning and configuration, users should have access to:

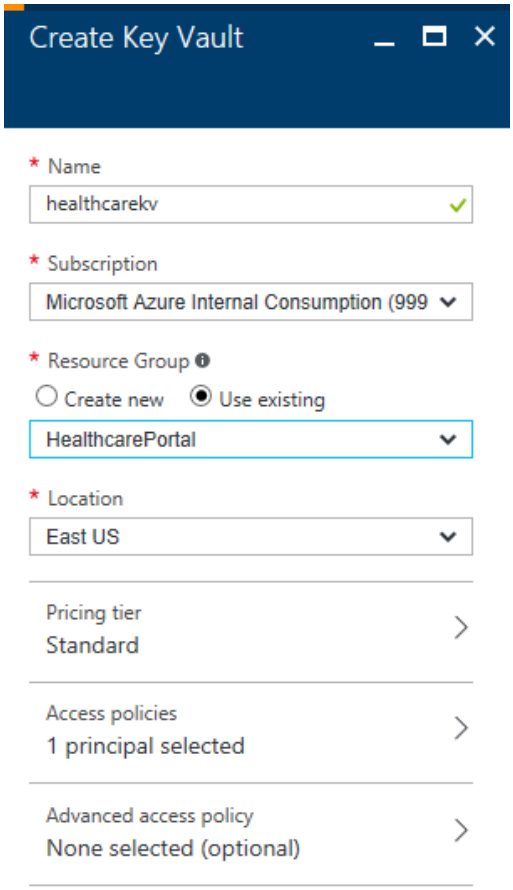
- Create an Active Directory application using PowerShell.
- Assign the service principal to an Azure Active Directory application.

4.5.1 Key Vault Provisioning

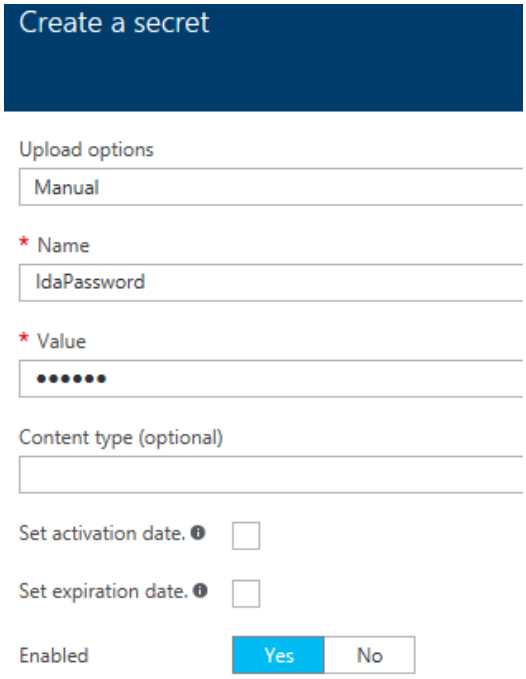
1. Login to Azure management portal.
2. Click on **New** (+) on the left navigation Panel.
3. Search for **Key Vault**.



4. Click on **Key Vaults**.
5. Click on **Create**.
6. Fill in the details as shown below.



7. Click on **Create.**
8. Once it is provisioned, open the Key Vault.
9. Click on **Secrets.**
10. Click on **Add** and fill the values like below.



11. Click **Create.**
12. Add another secret with Name "SpoPassword", "EncryptionKey" and "EncryptionSalt". The values of these secrets will be, for example,

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.

13. Note down the Key Vault Base URL in the format below. (The Key Vault Base URL is the DNS Name listed on the Properties page.)

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

4.5.2 Configure Azure AD Application for Key Vault and Associate the Certificate

1. 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.

2. Run the following in PowerShell after updating the text highlighted in yellow below, which creates the following items:
   - The Active Directory application with the certificate.
   - The service principal.
   - Assigns reader role to the service principal.

Below, replace the text highlighted in yellow with the 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

```
PS C:\WINDOWS\system32> Set-AzureSubscription -SubscriptionId "
```

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

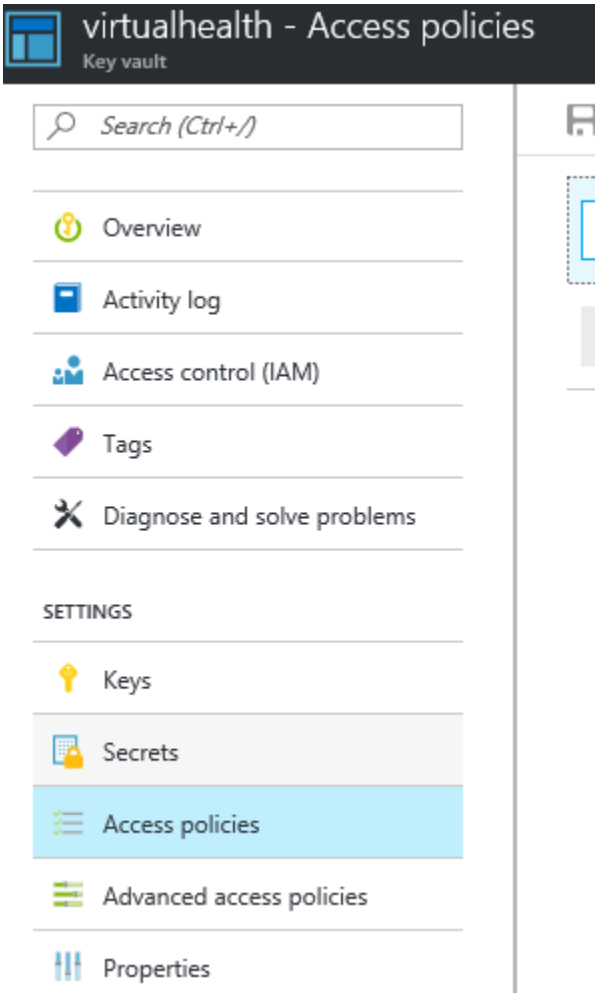
3. Note down the thumbprint and application Id as it will be used in Web.config of the HealthCare.Portal application.

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

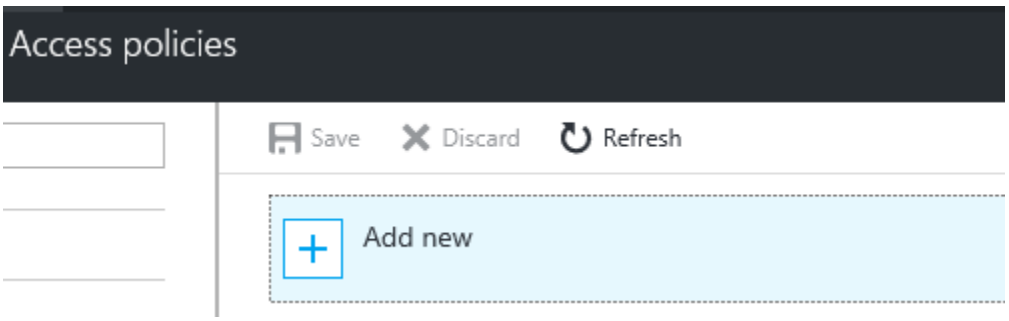
```
<add key="Thumbprint" value="THUMB_PRINT_CERTIFICATE" />
```

Add Access Policy to Key Vault

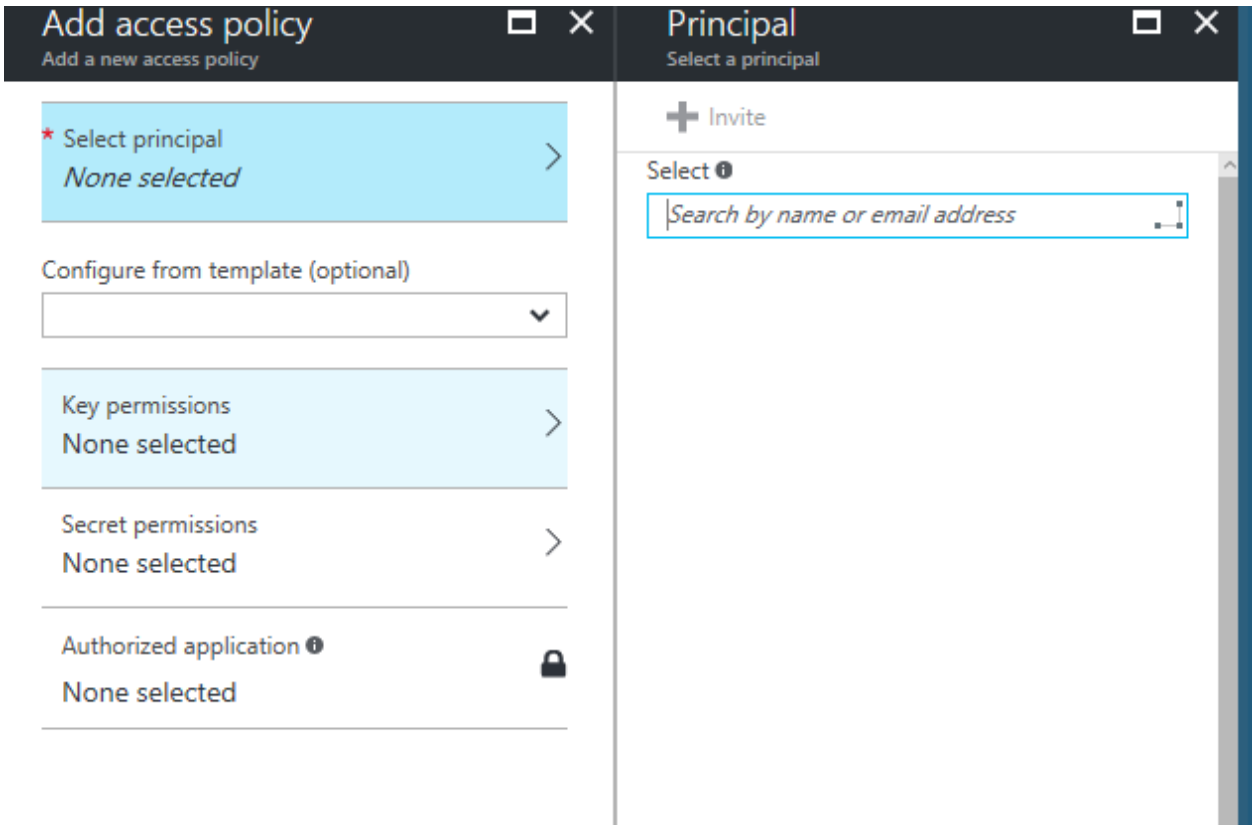
1. Go to the Azure portal.
2. Browse to **Key Vault** created earlier.
3. Go to **Access Policies.**



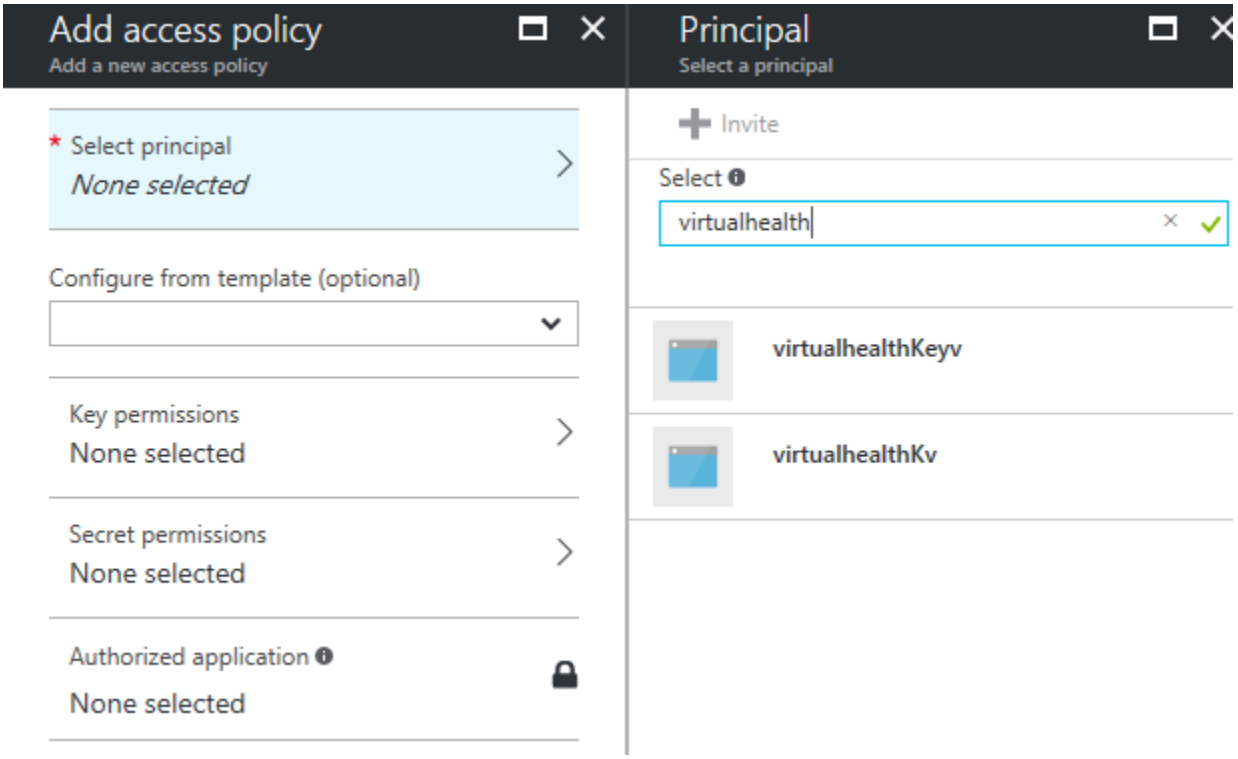
4. Click on **Add New.**



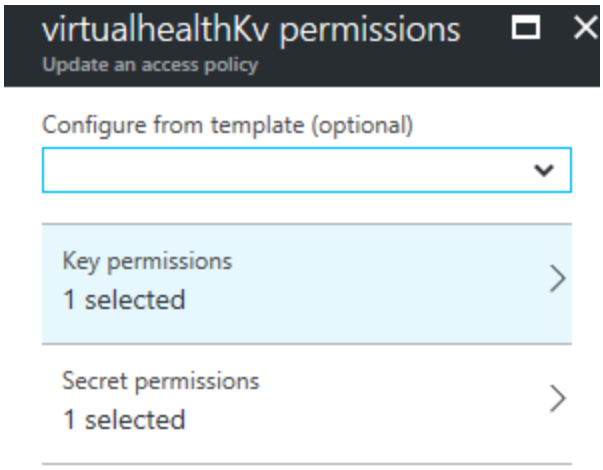
5. Click on **Service Principal.**



6. Search the principal that was created earlier such as virtualhealth.



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

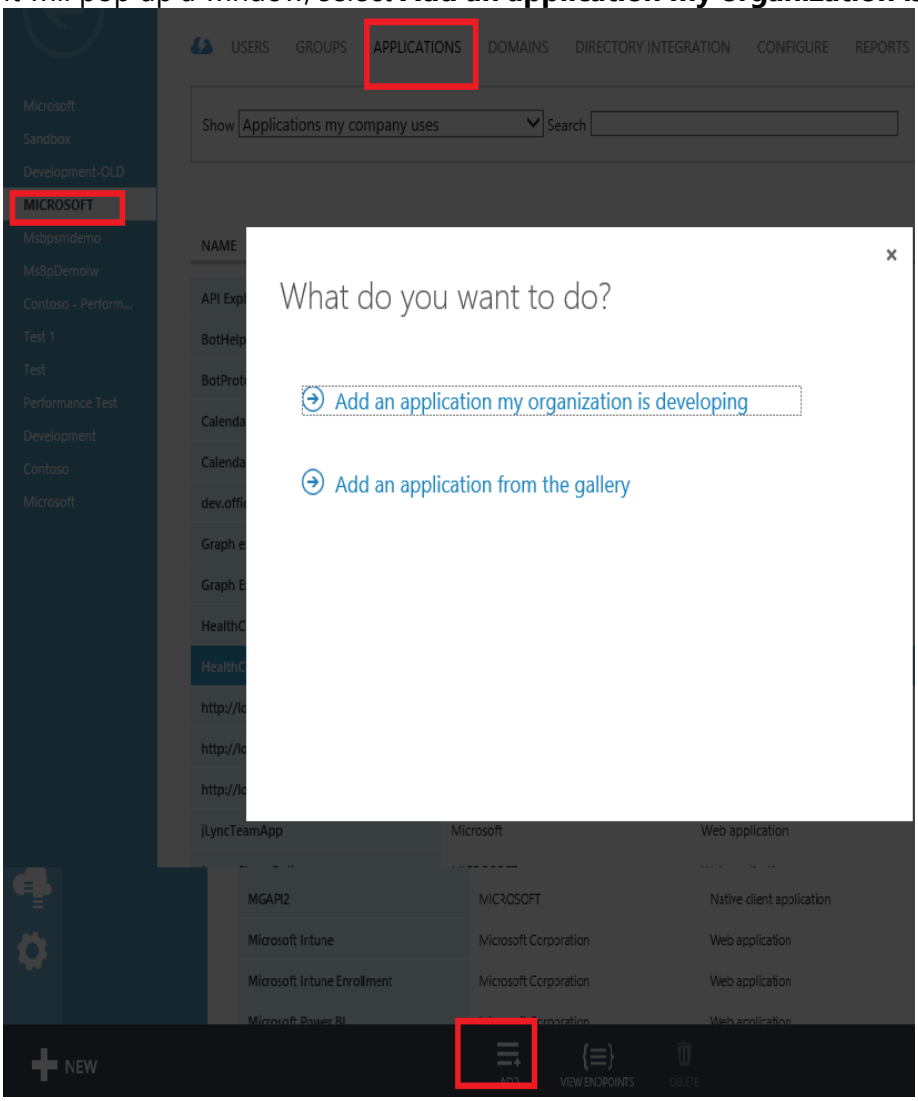

virtualhealthKv permissions
Update an access policy
Configure from template (optional)
Key permissions
1 selected
Secret permissions
1 selected

9. Click **Save**.

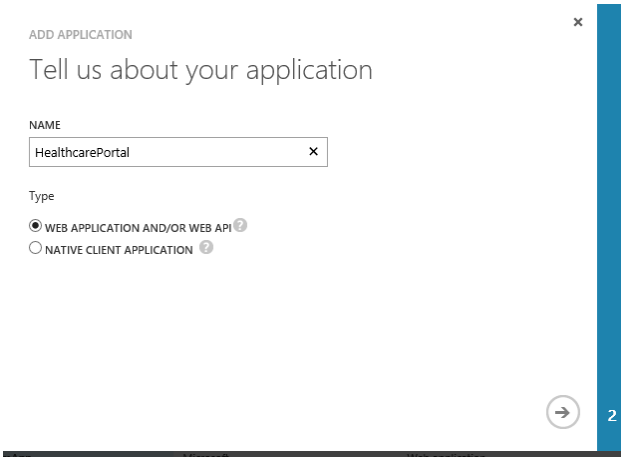
4.6 Azure Active Directory (Azure AD) Application Provisioning and Configuration

4.6.1 Provision Azure AD applications

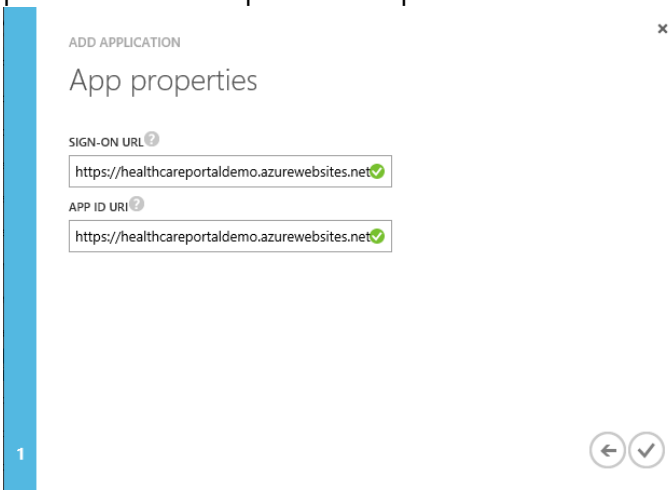
1. Login to [Classic Azure Management Portal] with Azure Admin Account.
2. Click on the **Active Directory** link on the left menu.
3. Select the **Active Directory**.
4. Click on **Applications** tab.
5. Then click on **Add link** at the bottom.
6. It will pop up a window, select **Add an application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



7. Provision Azure AD application for HealthCare Portal.
    a. Select the “Web Application AND/OR Web API” option.
    b. Provide the name “HealthcarePortal”.



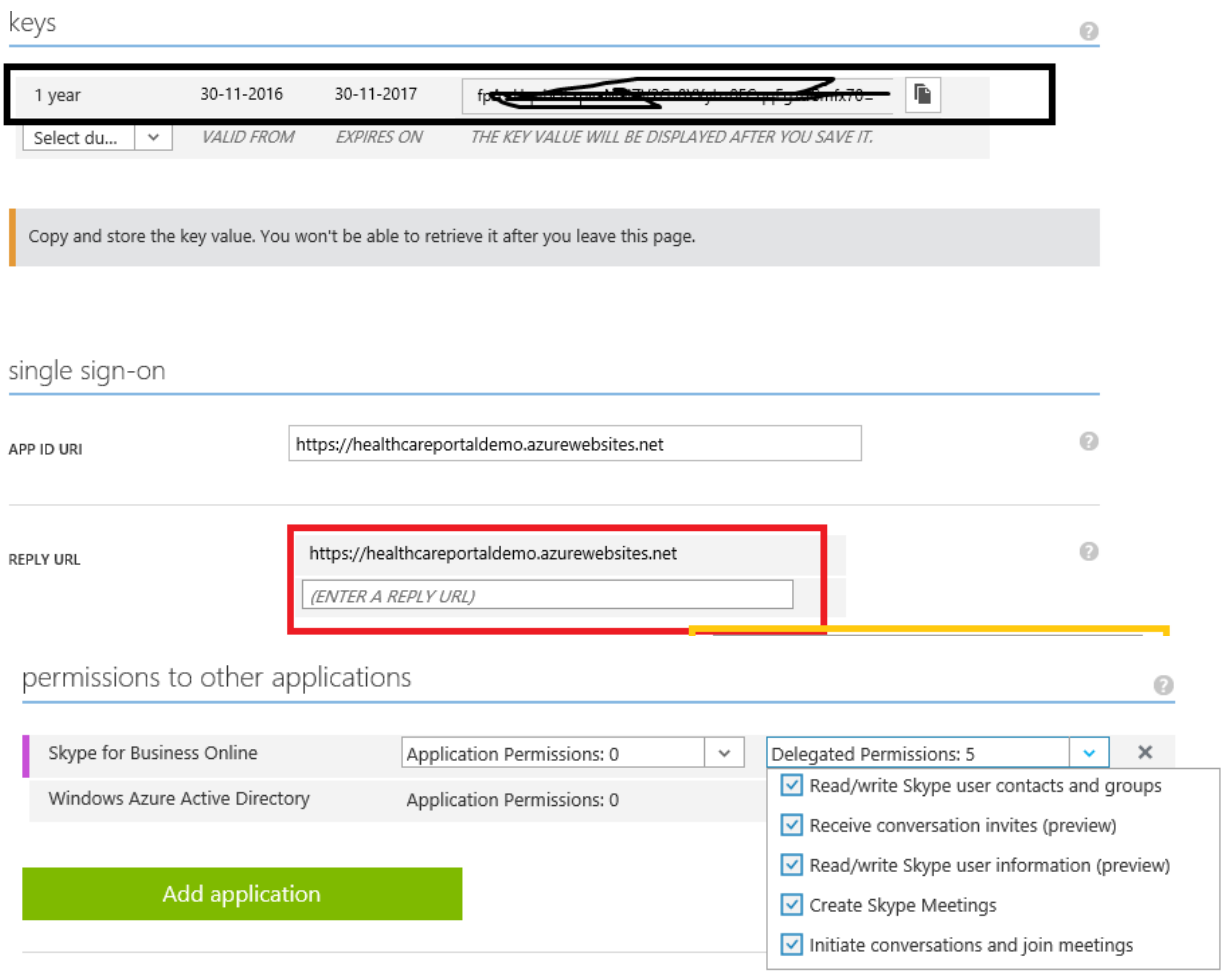
c. Click the arrow to go next.
d. Enter SIGN-ON and APP ID URL as the valid URL (e.g. the Azure portal Web App provisioned in the previous steps is shown as an example below).



e. Click the checkmark to okay**.**
f. Click on the created application.
g. Then navigate to the Configure tab.
h. Go to the key's section, add a key or select duration and save the application settings by clicking **Save** at the bottom of the screen.
i. It will generate a secret against the key. Make a note of it.
j. Make a note of the client Id.
k. Now go to the single sign-on section and add the Reply URL as per application URL (i.e. HealthCare.Portal URL).

   Note: The Reply URL MUST end with a forward slash, otherwise the user will get an ambiguous error when trying to log into the application.

l. Go to permissions to other applications.
m. Click on **Add application**.
n. Select **Skype for business online** and click the checkmark to okay**.**
o. Select the delegated permissions as shown in the image below.



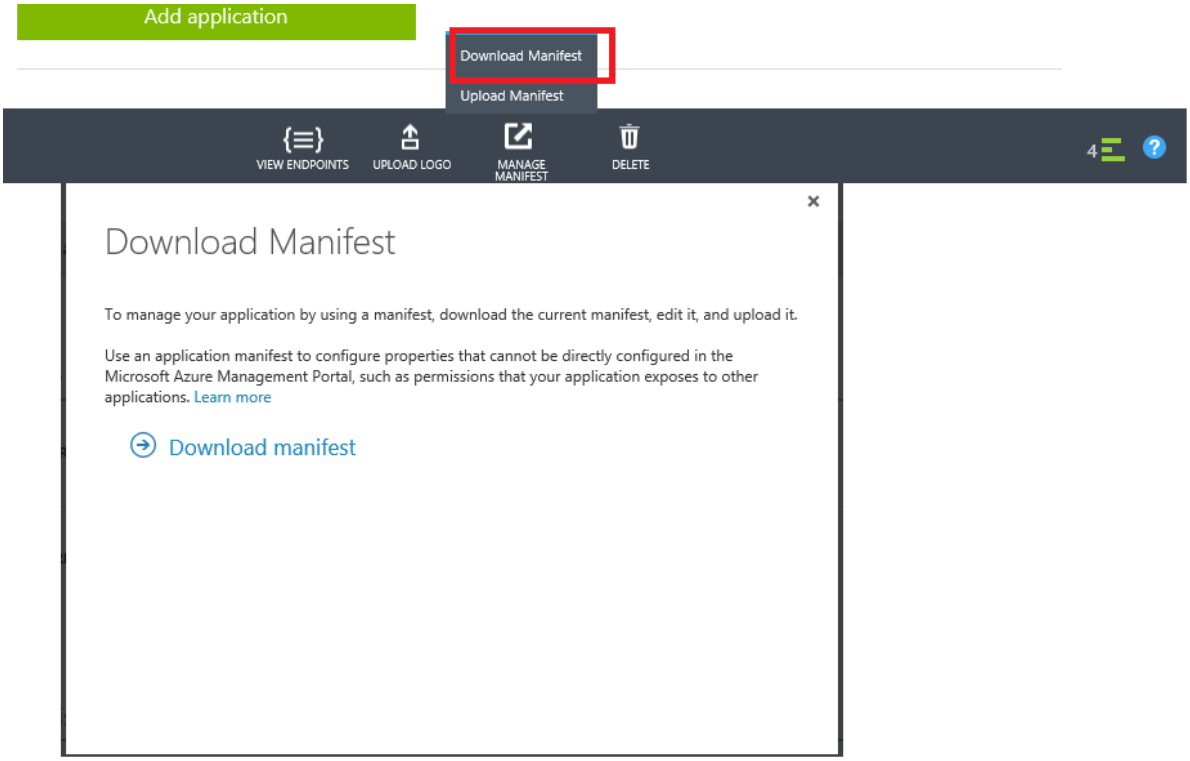
p. Click **Save** to save the application settings.

q. Note down the application secret from keys sections.

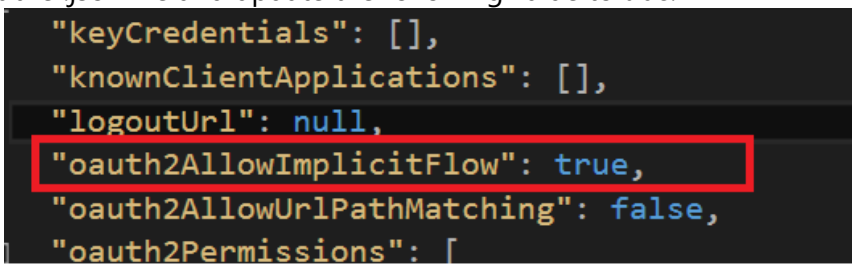
```
<add key="ida:ClientId" value="CLIENT_ID_AZURE_AD_APP_HEALTH_CARE_PORTAL" />

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

4.6.2 Configure OAuth Flow

1. Open the Azure AD application created in step 7.
2. Go to the **Configure** tab.
3. At the bottom, click on **Manage Manifest.**
4. Click on **Download Manifest.**



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



5. Save the file.
6. 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. Go to portal.azure.com to make configurations.

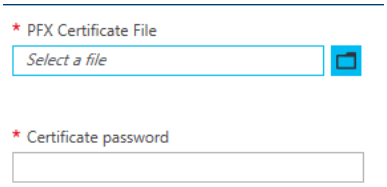
4.7.1 Add the Certificate to Web App

1. Click on **App Services**, available in the left navigation of the Azure portal.
2. Click on **Web App**.
3. Click on **SSL Certificates**.
4. Click to upload the certificate.

   Note:
   If you are using a self-signed certificate, then this certificate is the same certificate which was created during the Key Vault provisioning, section 4.5. Export the *.pfx file with the password.

   If you are using the CA certificate, then you should have a *.pfx file available to you.

5. Select *.pfx file and enter the password for the PFX file.



6. Click **Upload.**

4.7.2 Add the Key to Azure Web App

To load the certificate, add the following entry to Azure Web App.

7. Click on **Application Settings**.
8. Scroll down and look for the App Settings section.
9. Enter WEBSITE_LOAD_CERTIFICATES as key and * as value.
10. Save the setting.

4.8 Other Configurations

This section lists other configurations elements that need to be captured before deploying the web apps. For example, the tenant ID can be obtained in section 4.5.2 after the user executes the Add-AzureRmAccount command in PowerShell. It will be displayed on the screen with a few other items.

```
<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.

Note: If the project is loaded with Visual Studio 2017 and you see the following errors for the HealthCare.Core and HealthCare.Portal projects:

```
Error encountered while loading the project. Some project features, such as full solution
analysis for the failed project and projects that depend on it, have been disabled.

This project references NuGet package(s) that are missing on this computer. Use NuGet Package
Restore to download them.  For more information, see
http://go.microsoft.com/fwlink/?LinkID=322105.
```

Fix this by selecting **Tools** -> **NuGet Package Manager** -> **Package Manager Console**, click on the **Restore** button.

After fixing the NuGet package issue and building the project, 305 warnings may be generated. This, however, does not interfere with the publishing process and can be ignored.

5.1.1 Update Bot Web.config

1. Click on the VHCBot project and expand it.
2. Open the Web.config file.
3. Update the following values.

```
<add key="BotId" value="BOT_ID" />
<add key="MicrosoftAppId" value=" BOT_APP_ID" />
<add key="MicrosoftAppPassword" value="BOT_APP_PASSWORD" />
<add key="ida:HealthCarePortal" value="HEALTH_CARE_PORTAL_URL"/>
<add key="ida:UserName" value="O365_USER_NAME" />
```

4. Save and close the file.

5.1.2 Update the HealthCare Portal URL

1. Click on the HealthCare.Portal project and expand it.
2. Open the Web.config file.
3. Update the following values with values captured in the above section (use your notepad or xml notepad). For example:

   The key TrustedApi has the value of the URL of the service that is deployed as the Trusted Application Endpoint.

   The key SharepointAdminGroup has the value of "VirtualPatientCare Admin" as per section 4.2.

```
<!--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" />
```

4. Save and close the file.

5.1.3 Update Deployment Tool Config file

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

1. Open the DeploymentTool.exe.config file after extracting the contents of the VirtualHealthDeploymentTool.zip 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"/>
```

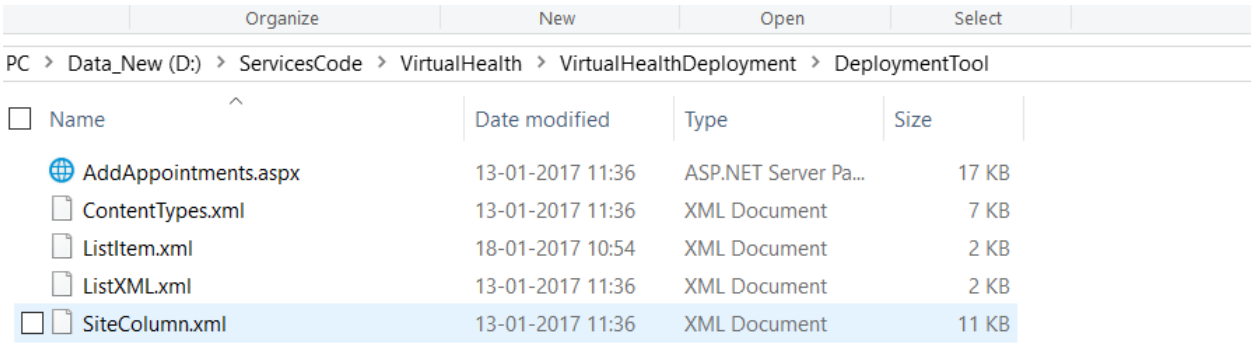
3. Save it.

5.2 SharePoint Deployment

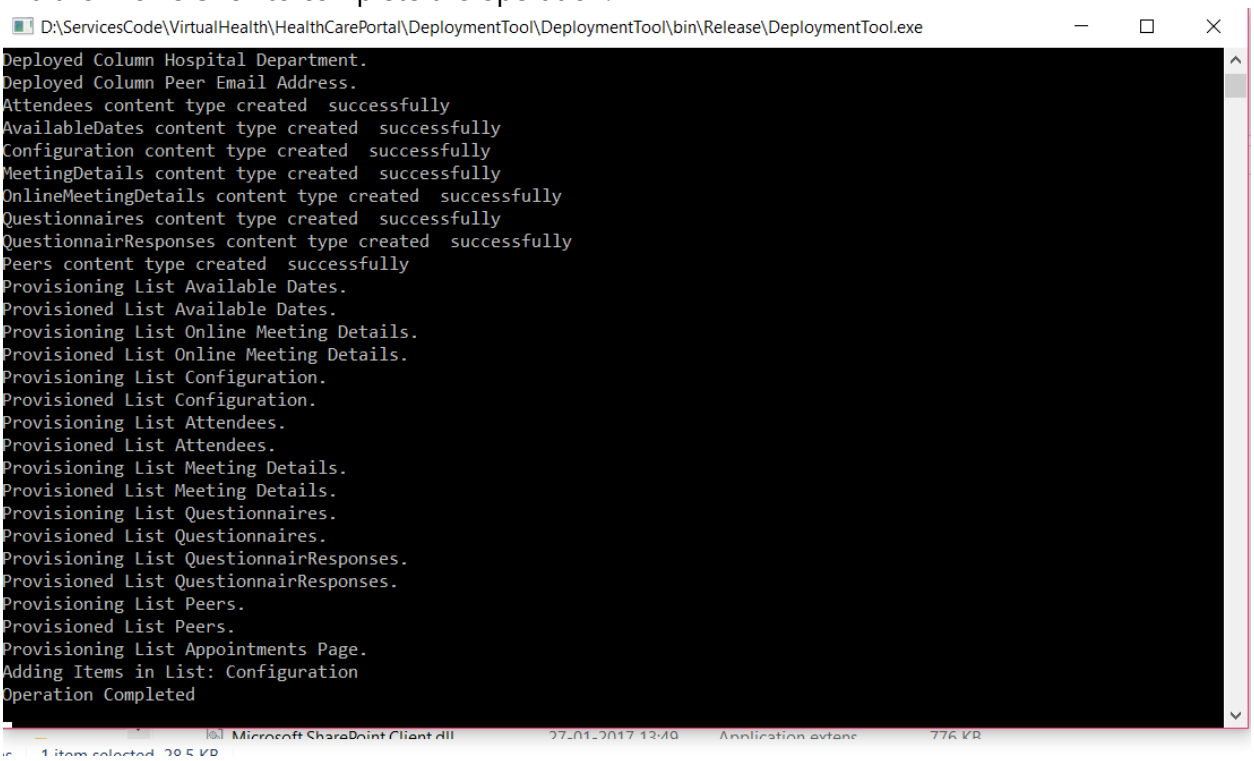
Deploy the SharePoint artifacts using thesteps below:

1. Download the deployment tool .
2. The folder contains the 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



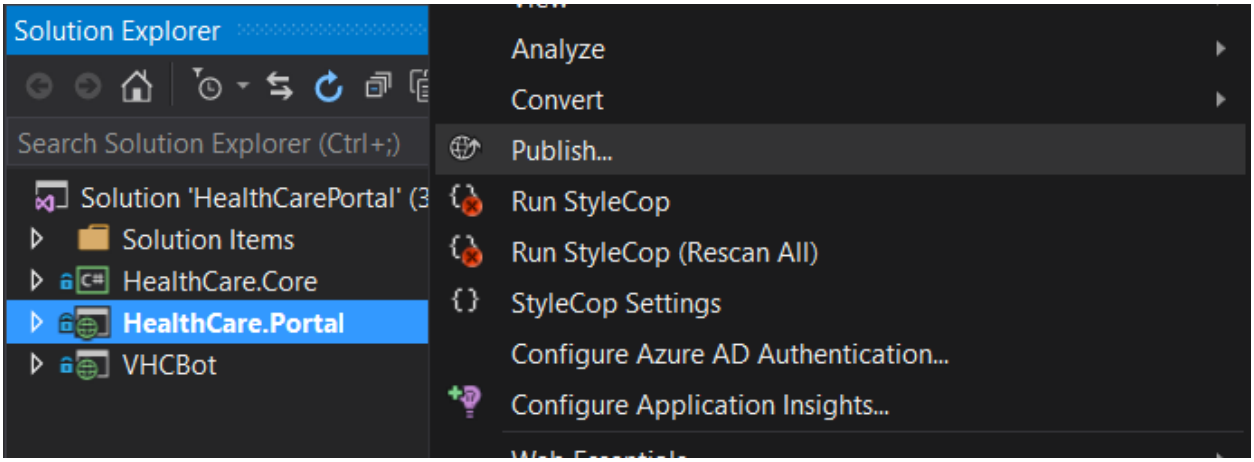
3. Double click **DeploymentTool.exe**
4. Select **Option 6.**
5. Wait for PowerShell to complete the operation.



5.3 Virtual Health Deployment

There are multiple ways to deploy a web app in Azure. This section is about Visual Studio Web Deploy publishing. Note that the UI shown in the following steps may differ from the latest Visual Studio 2017.

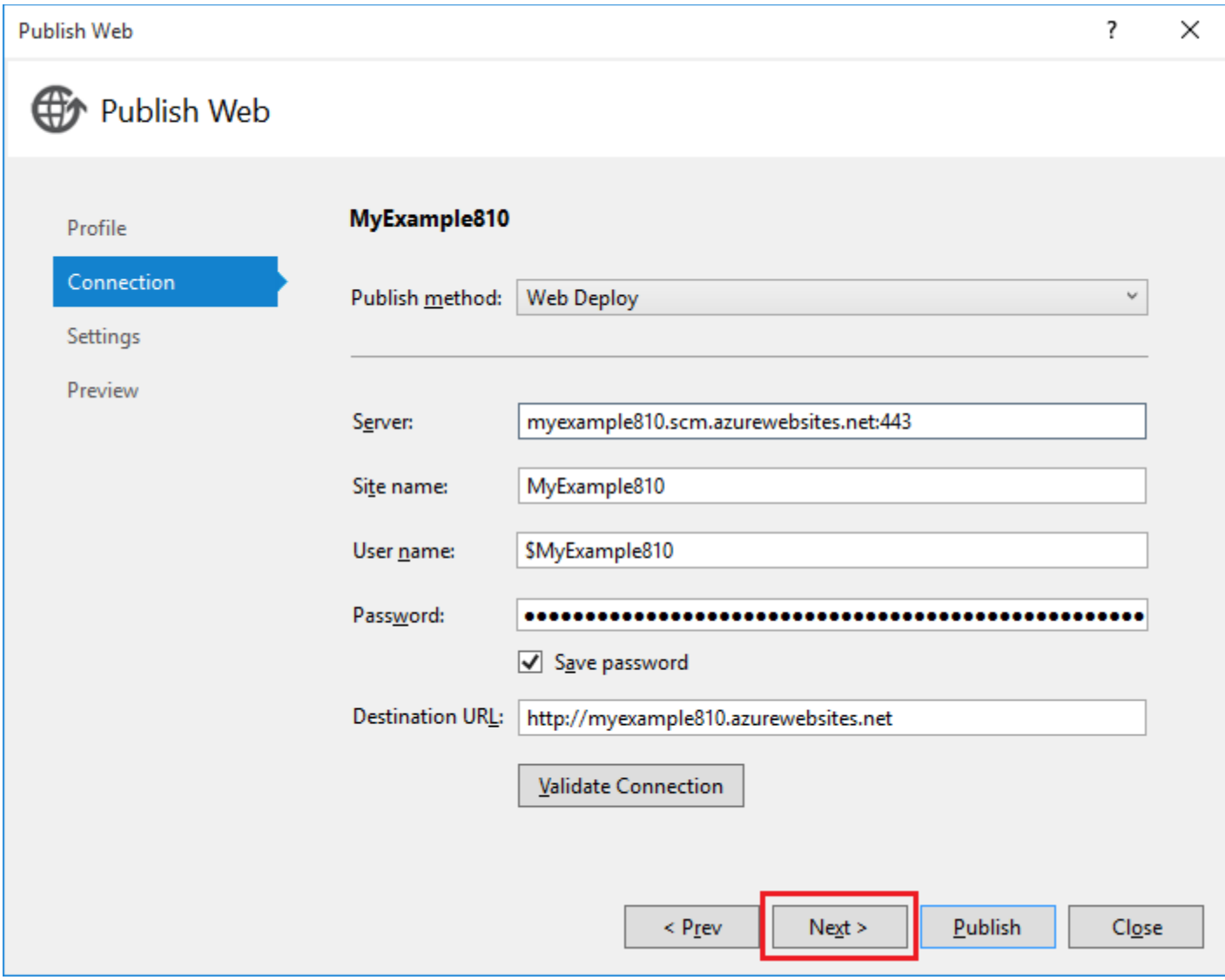
In **Solution Explorer**, right-click the project, and choose **Publish**.



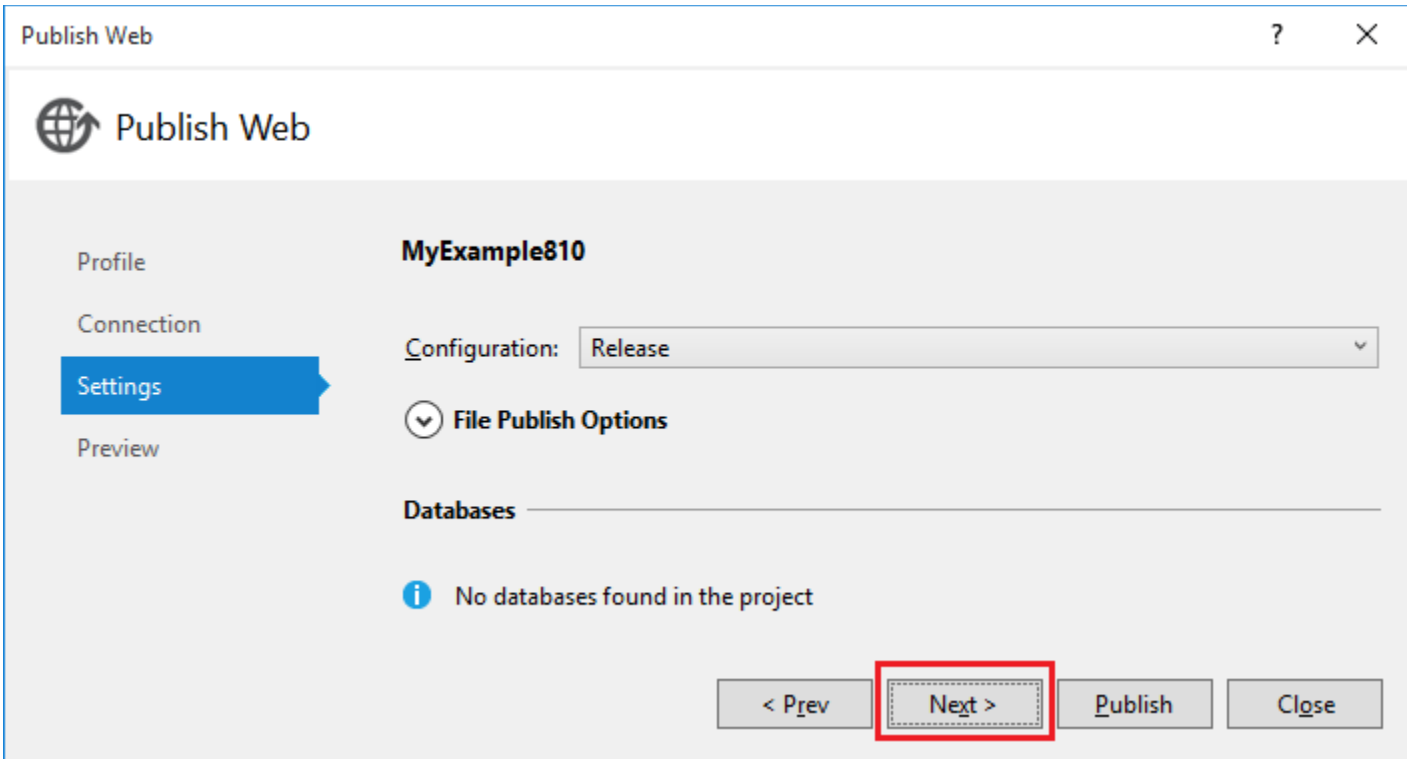
In a few seconds, the **Publish Web** wizard appears. The wizard opens to a *publish profile* that has settings for deploying the web project to the new web app.

> 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**.

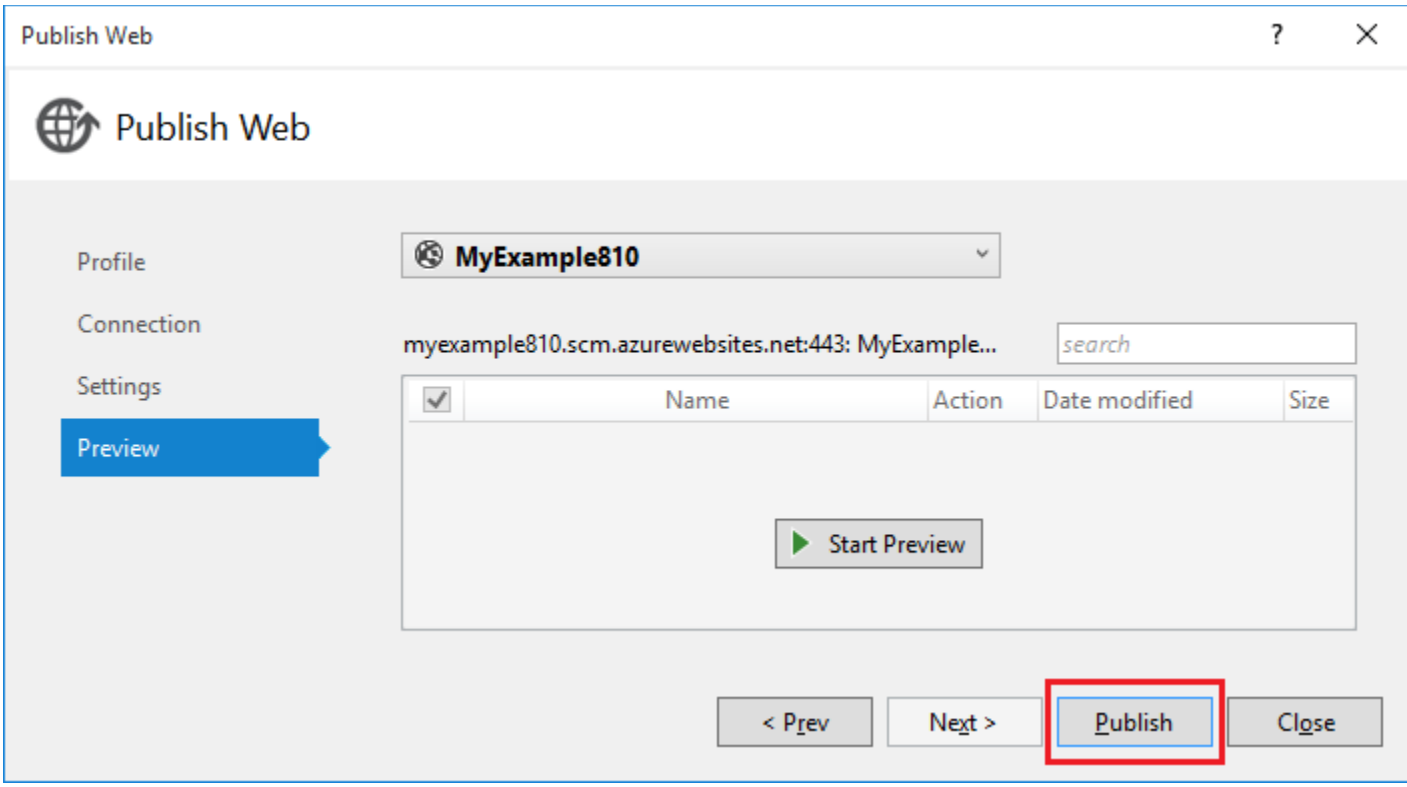


2. The **Settings** page is where you can change the build configuration to deploy a debug build for remote debugging. **Settings** also offers several **File Publish Options**.



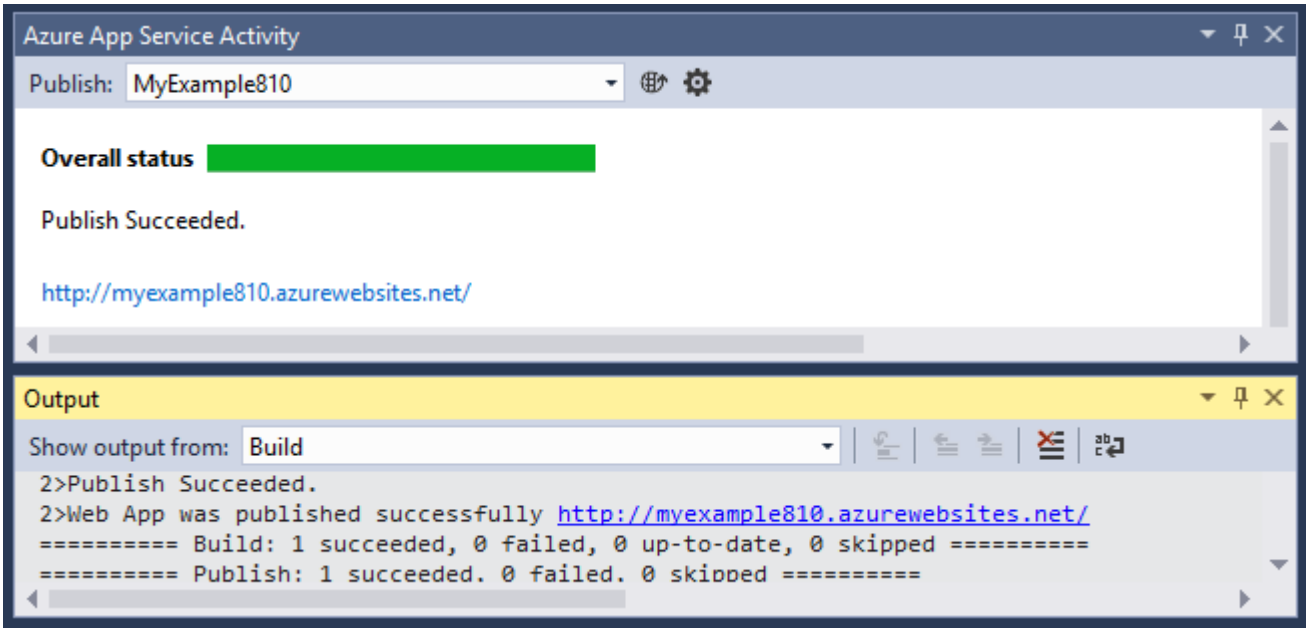
On the **Settings** tab, click **Next**.

3. The **Preview** page shows the files that are going to be uploaded from your project to the API app. After a project is deployed, you may need to redeploy a project with updates to an API app. Only changed files are uploaded. To see a list of files to be published, click the **Start Preview** button.



On the **Preview** page, click **Publish** to begin the process of uploading the files to the Azure server. This may take a minute or two.

The **Output** and **Azure App Service Activity** windows show the deployment actions that were taken and whether the deployment was successful.



Upon successful deployment, the default browser automatically opens to the URL of the deployed web app. 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 VHC Bot Web App Deployment

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

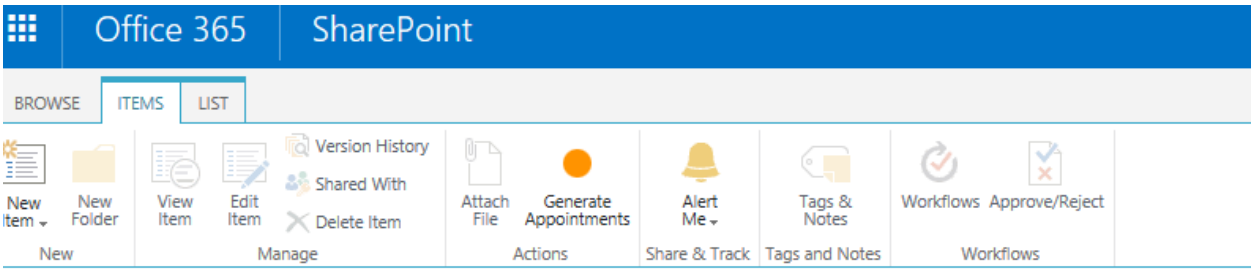
5.5 Post Deployment Configuration

5.5.1 BotUrlEmbed in Web.config file of HealthCare.Portal

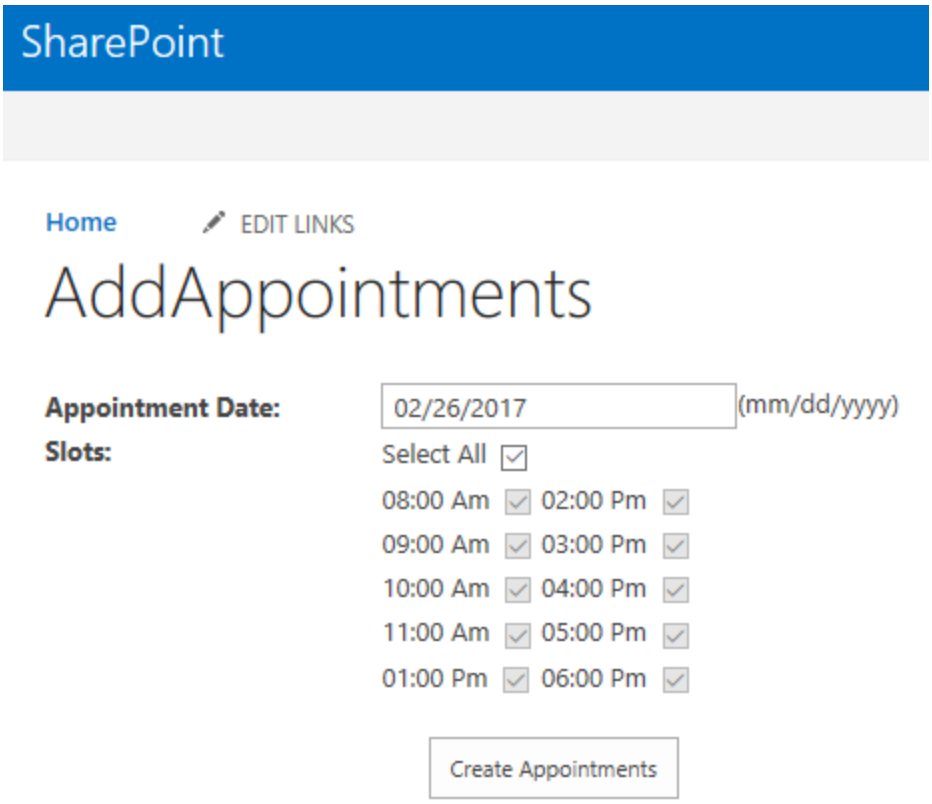
1. Go to section 4.4 and Get URL Embed code.
2. Note down the BotUrlEmbed.
3. Update the Web.config of HealthCare.Portal with the above value.
4. Republish the HealthCare.Portal project again using Visual Studio.

5.5.2 Generate the Meeting Slots

1. Open the SharePoint Site Collection created in section 4.1.
2. Login as site collection administrator.
3. Go to **Settings → Site Contents**
4. Click on the **Available Dates** list
5. On the bottom left corner, there will be a link to change to the classic view, click on **Return to classic SharePoint**.
6. Click on the **Items** tab and then click on **Generate Appointments.**



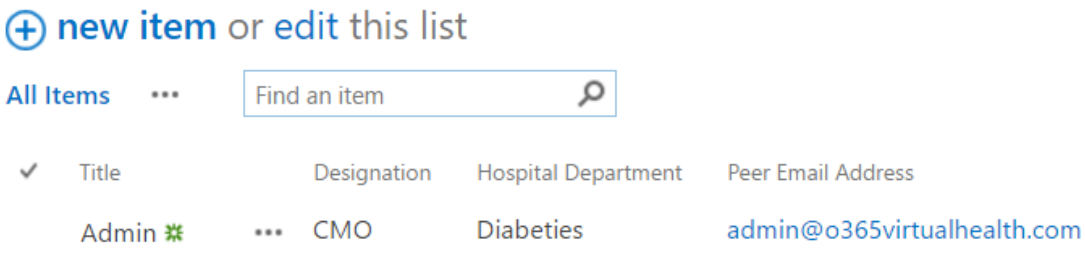
7. Fill in the details as shown below.



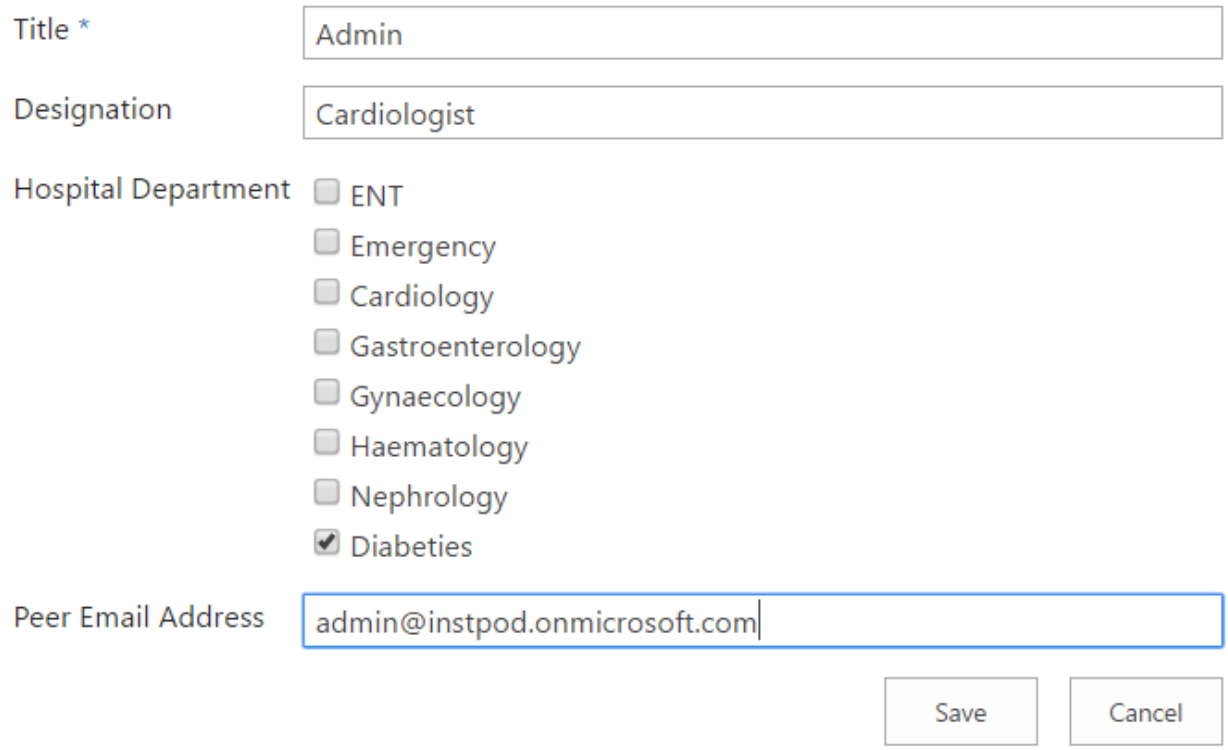
8. Click **Create Appointments** to see the confirmation message.

5.5.3 Populate Peers List

1. Open the SharePoint Site Collection created in section 4.1.
2. Login as the site collection administrator.
3. Go to **Settings → Site Contents.**
4. Click on the **Peers** list.
5. On the bottom left corner, there will be a link to change to the classic view, click on **Return to classic SharePoint.**
6. Click on **new item.**

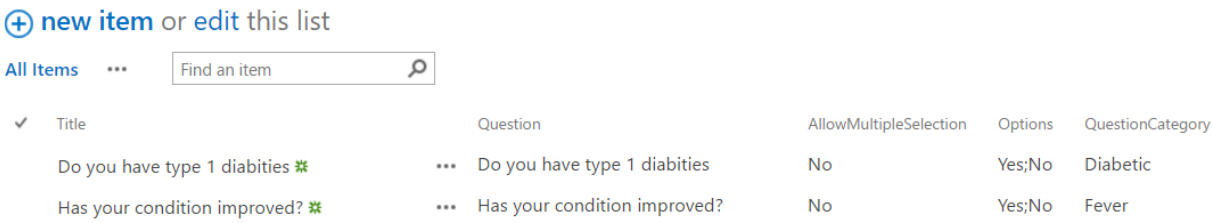


7. Add details of admin and other doctors as below and click on **Save**.

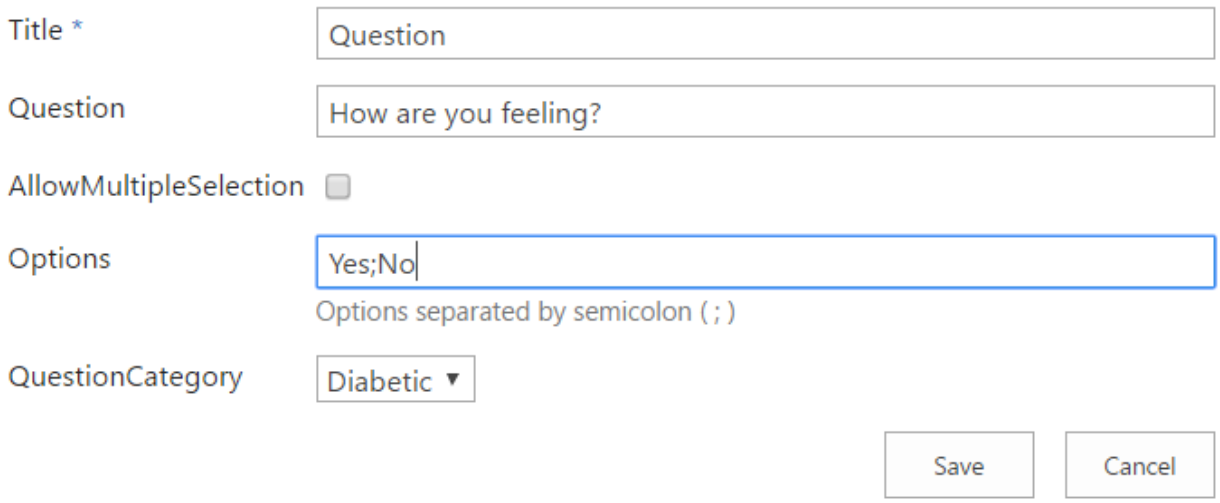


5.5.4 Populate Questionnaire List

1. Open the SharePoint Site Collection created in section 4.1.
2. Login as the site collection administrator.
3. Go to **Settings → Site Contents.**
4. Click on the **Questionnaires** list.
5. On the bottom left corner, there will be a link to change to the classic view, click **Return to classic SharePoint**.
6. Click on **new item**.



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



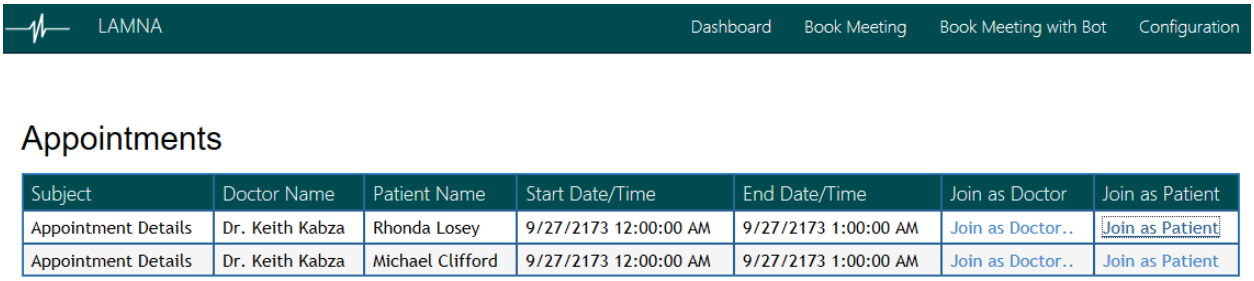
6 Post Deployment Validations

6.1 Validate Home Page is Opening

1. Open the HealthCare Portal Web App.
2. Login as an Active O365 Tenant user. The Home page looks like below when there is no appointment.

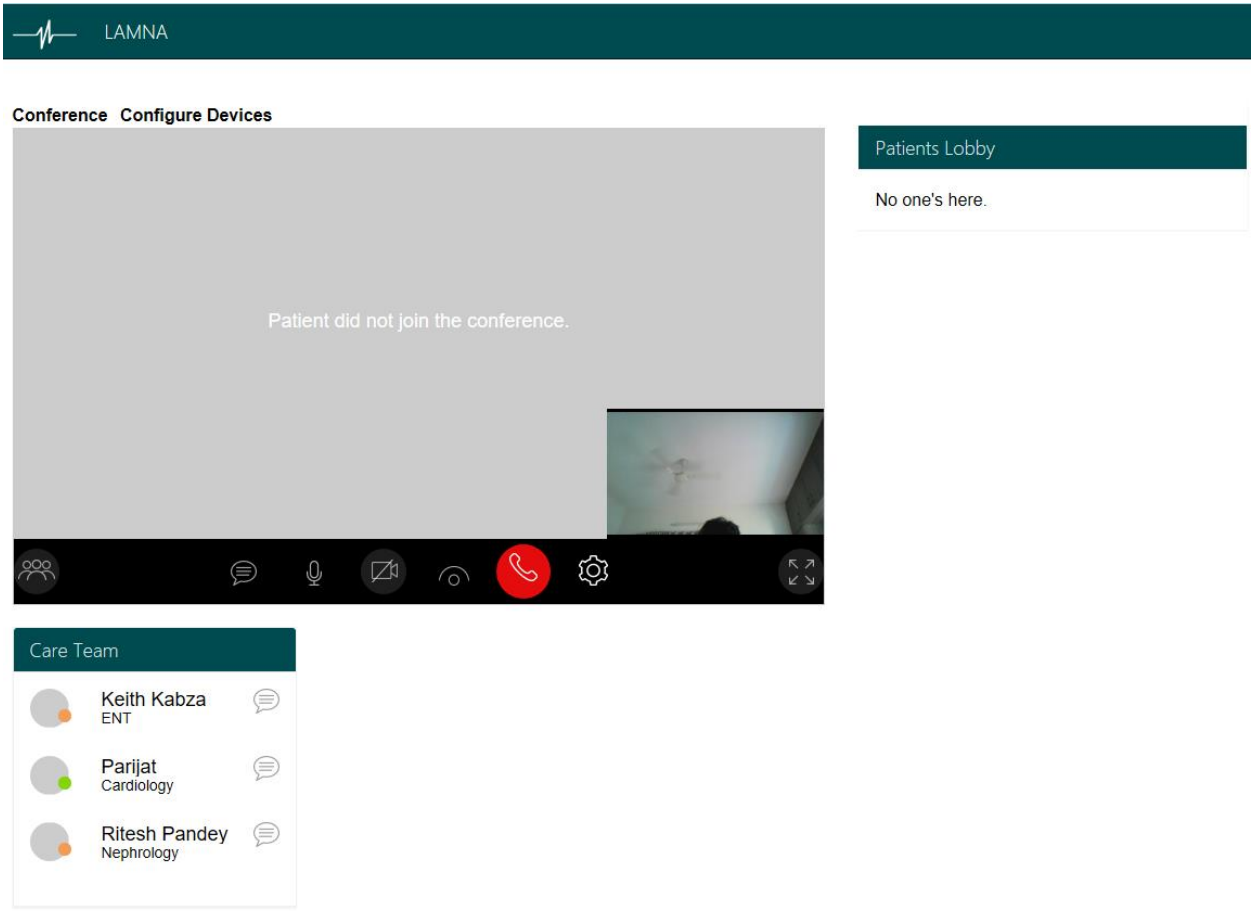
6.2 Validate Book Meeting

1. Click on the **Book Meeting** link on the top navigation.
2. Fill in the details.
3. Click **Submit**. The dashboard shows the meeting in a grid as shown below.



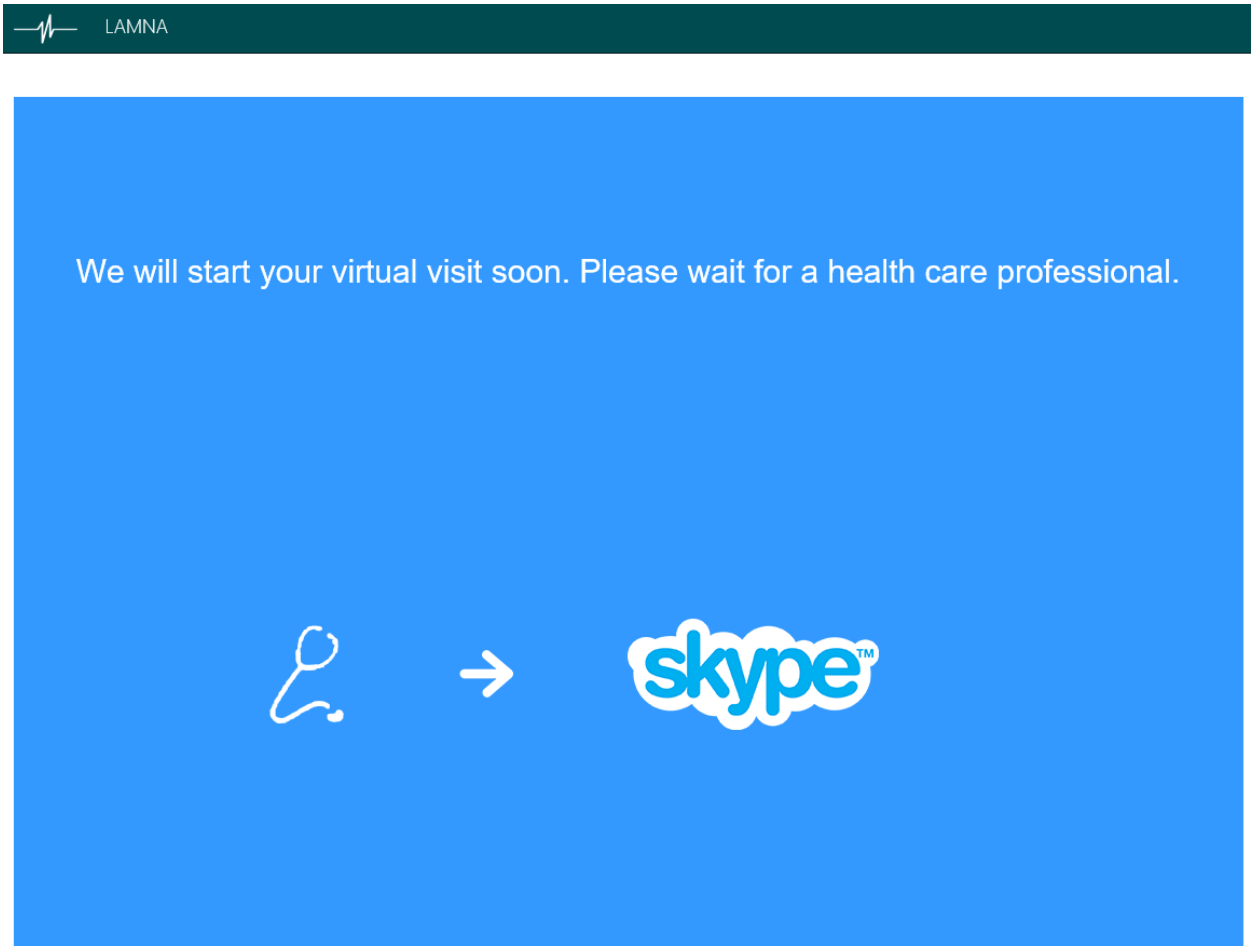
6.3 Join a Conference as Doctor

1. On the Home page, click on the **Join as Doctor** link to open the page in a new tab.
2. Wait for the page to load. See the image below.

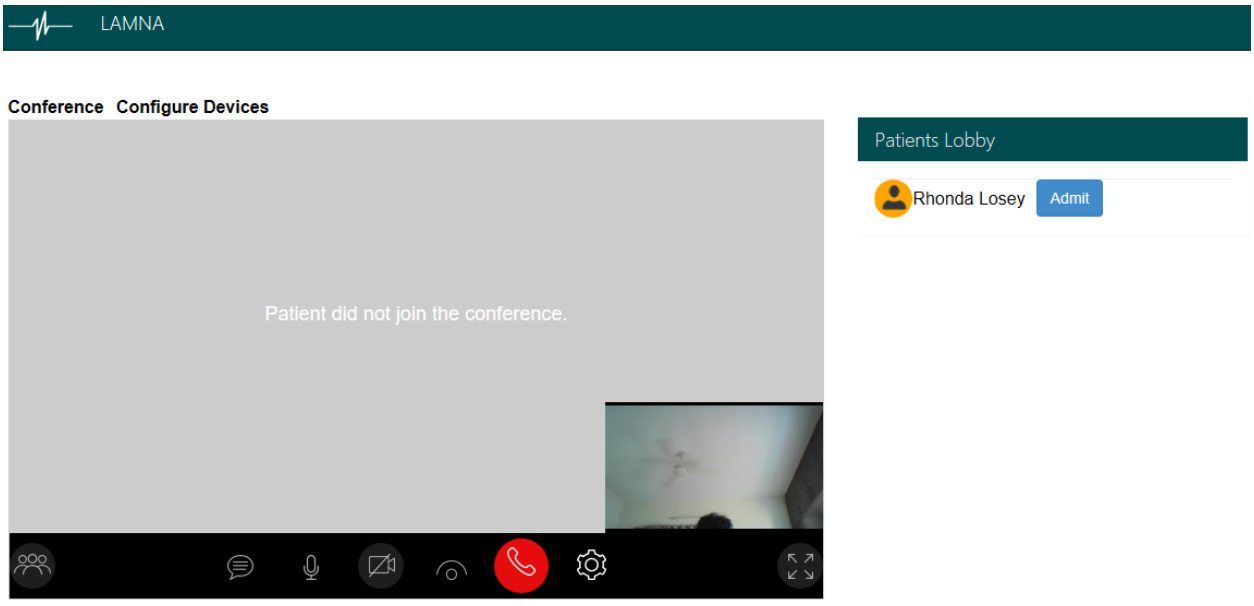


6.4 Join a Conference as Patient

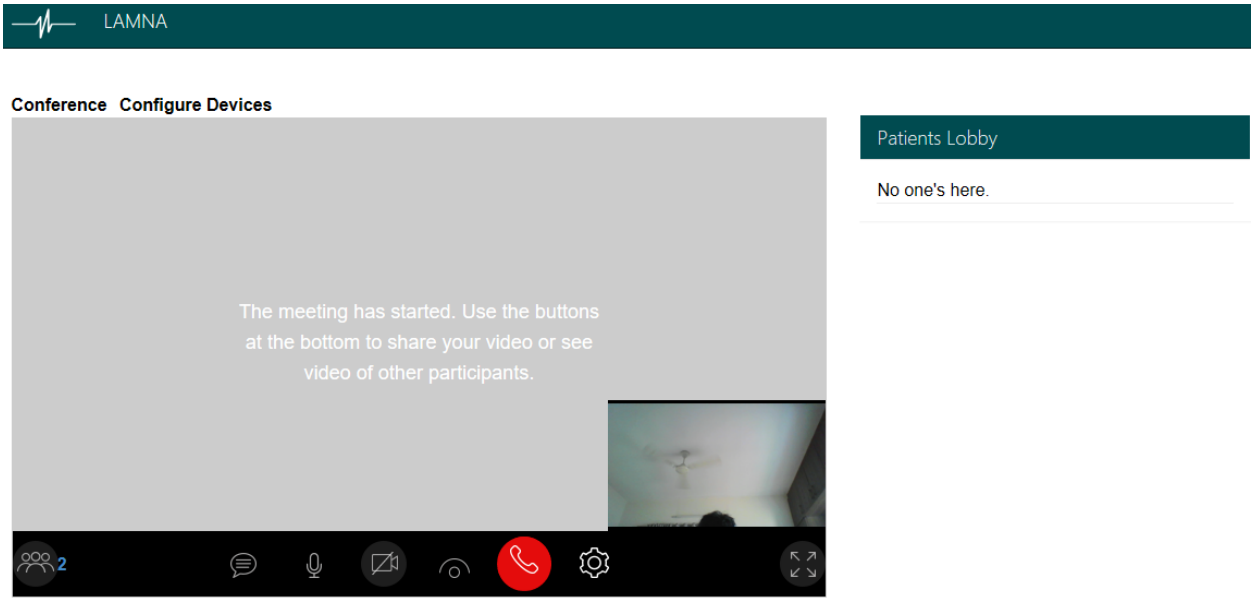
1. Go to the dashboard.
2. Copy the URL of the patient link from the meeting that the doctor has joined earlier.
3. Open a new browser window and paste the URL.
4. Wait for the page to load. The patient will wait in the lobby.



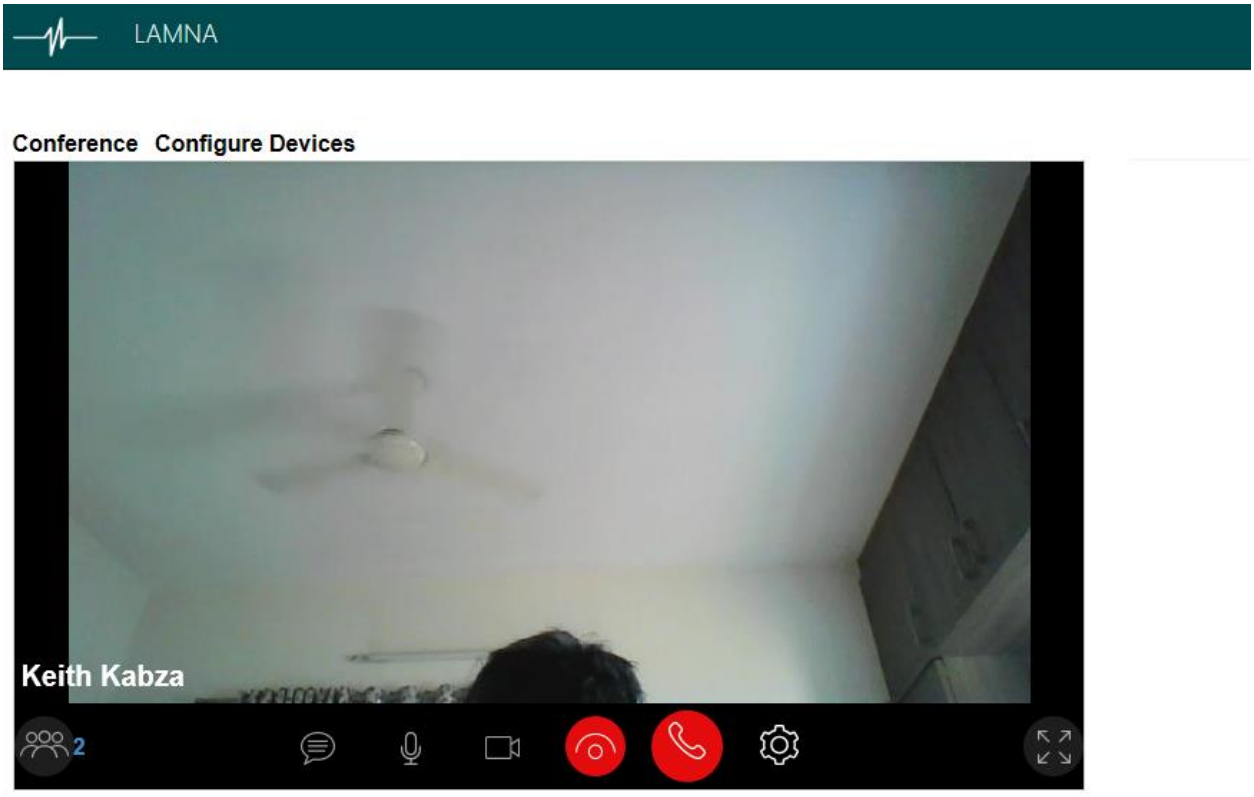
5. Go to the doctor's window.
6. Click on **Admit.**



7. Meeting starts. The doctor's window is shown below.

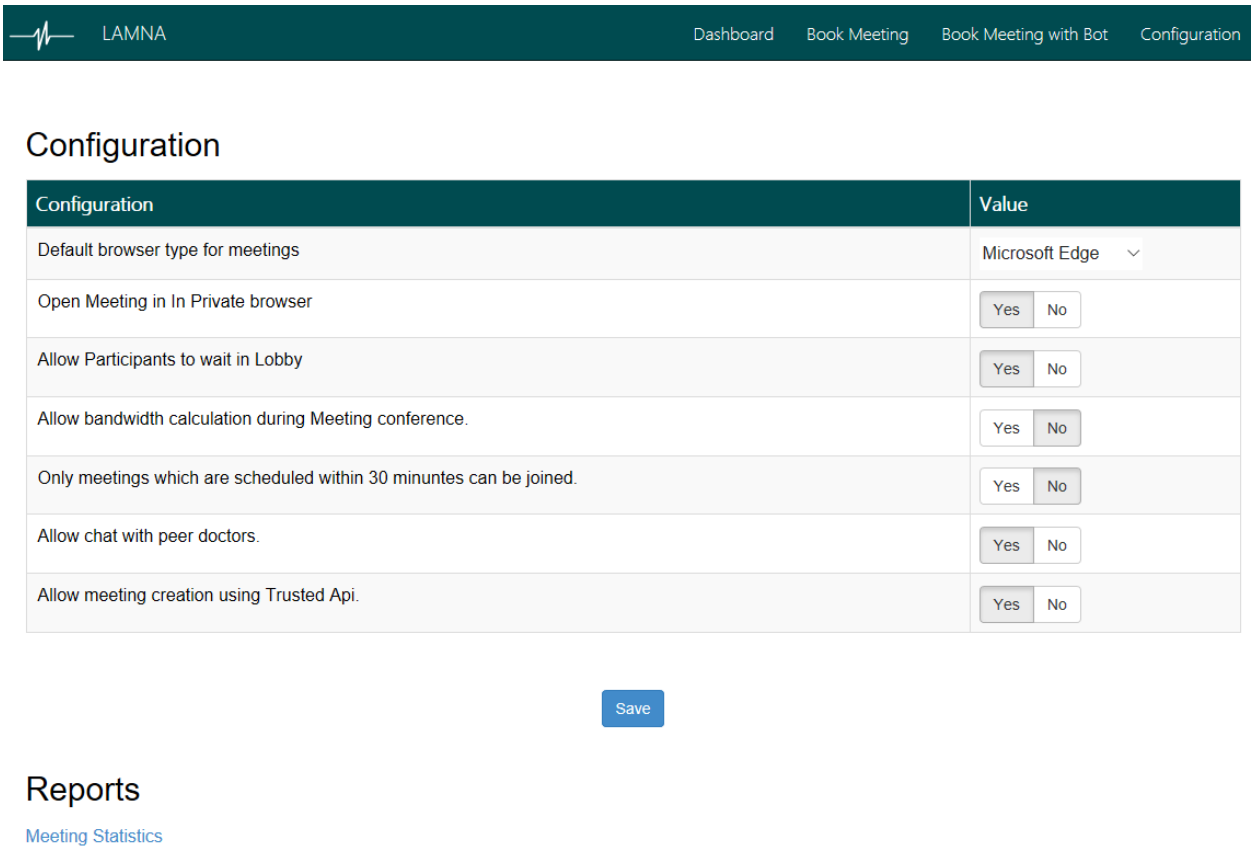


8. Go to the patient's window, shown below.



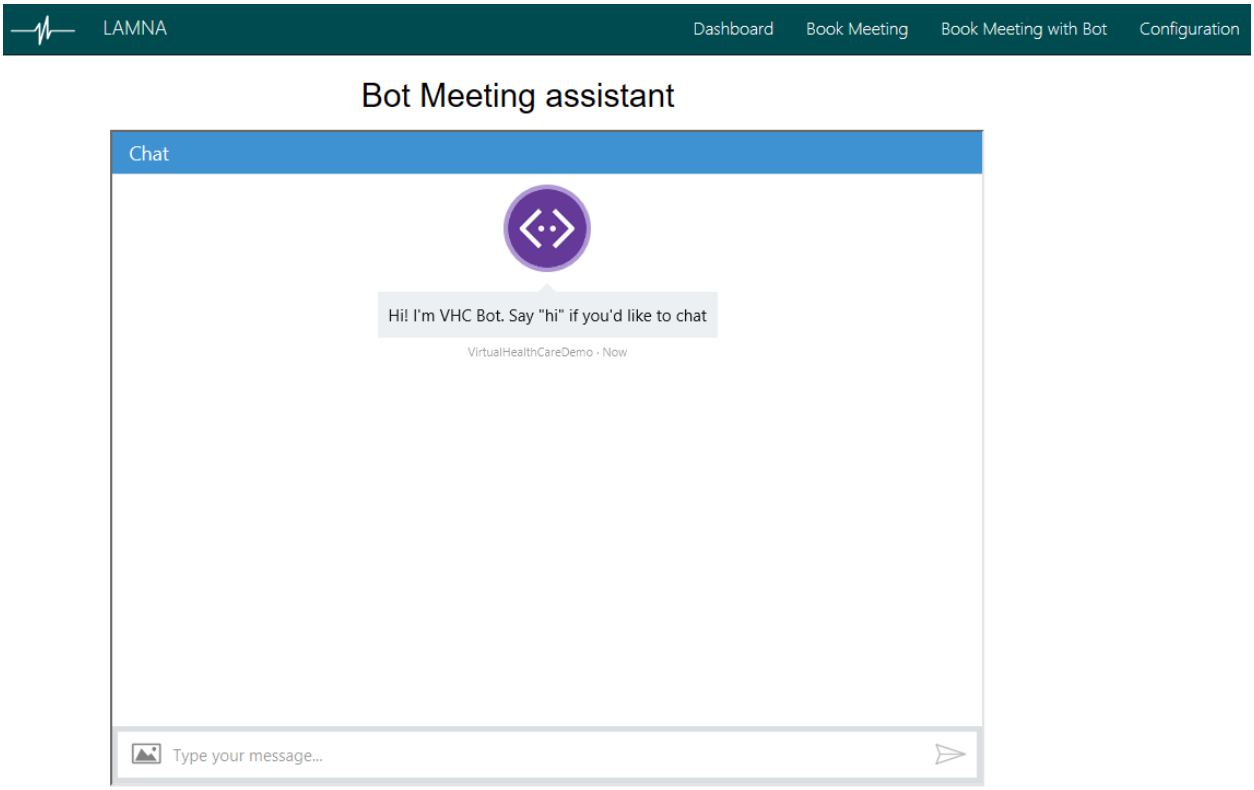
6.5 Verify Configuration Page

1. Click on the **Configuration** link in the navigation.
   If the user is a member of "**VirtualPatientCare Admin**", then the page will load and the user will see this page as shown below. Otherwise, the user will see an unauthorized message on the page.



6.6 Validate Book Meeting with Bot Page

Click on the **Book meeting with Bot** link in the global navigation to load the Bot page. See the image below.



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