# Step-by-Step Guide: Create Static Website and Add Chatbot using Microsoft Azure

## 1. Create an Azure Storage Account

1. Go to https://portal.azure.com and sign in.  
2. Click 'Create a resource' → search 'Storage account' → click 'Create'.  
3. In the Basics tab:  
 - Subscription: Choose your active subscription.  
 - Resource group: Create or select one.  
 - Storage account name: Must be unique and lowercase.  
 - Region: Choose your nearest region (e.g., Central India).  
 - Performance: Select 'Standard'.  
 - Redundancy: Choose from LRS, ZRS, GRS, GZRS.  
4. In Data protection: Disable Soft Delete.  
5. Click 'Review + Create' → 'Create'.

## 2. Enable Static Website and Upload Files

1. After creation, go to the storage account.  
2. In the left panel, under Data Management, select 'Static website'.  
3. Enable 'Static website'.  
 - Index document name: index.html  
 - Error document path: 404.html (optional)  
4. Click 'Save'. Azure creates a $web container automatically.  
5. Note the 'Primary endpoint' URL (this is your website link).  
6. Go to Containers → $web → Upload → Browse for files → Upload your website files (HTML, CSS, JS).  
7. Visit your 'Primary endpoint' URL to verify your static site.

## 3. Create a Custom Question Answering Resource (Language Service)

1. In Azure portal, click 'Create a resource' → search 'Language' → select 'Create'.  
2. Choose 'Custom Question Answering' under AI Services.  
3. Fill details:  
 - Resource Group: Same as your storage account.  
 - Region: Central India.  
 - Name: lang-<yourname>.  
 - Pricing Tier: Free.  
 - Azure Search Region: Central India.  
4. Tick the required checkboxes and click 'Create'.

## 4. Create and Configure a Project in Language Studio

1. Go to https://language.microsoft.com → Sign in.  
2. Click 'Get started with Language Studio' → Select your Azure resource.  
3. Click 'Create new project' → choose 'Custom Question Answering'.  
4. Enter:  
 - Project name  
 - Description  
 - Language  
 - Error message display settings  
5. Click 'Manage' → 'Add source' → 'File' → upload your FAQ or question-answer file.  
6. Click 'Save', then 'Train' the project and 'Publish' it.

## 5. Create Azure Bot and Link QnA Resource

1. Go to 'Create a resource' → search 'Azure Bot' → click 'Create'.  
2. Fill details:  
 - Resource Group: Same as others.  
 - Bot name: unique.  
 - Region: same as your language resource.  
3. Click 'Review + Create' → 'Create'.  
4. After creation, open your bot → in 'Configuration' link your QnA (Language) project.  
5. Click 'Channels' → select 'Web Chat'.  
6. Copy the Web Chat 'Secret' or 'Embed code' (for testing).

## 6. Embed the Chatbot in Your Static Website

1. Open your 'index.html' file.  
2. Paste the following code where you want the chatbot to appear:

<script src="https://cdn.botframework.com/botframework-webchat/latest/webchat.js"></script>  
<div id="webchat" role="main"></div>  
<script>  
 (async function () {  
 const directLineSecret = 'YOUR\_WEBCHAT\_SECRET\_FROM\_AZURE';  
 const res = await fetch('https://directline.botframework.com/v3/directline/tokens/generate', {  
 method: 'POST',  
 headers: { Authorization: 'Bearer ' + directLineSecret }  
 });  
 const { token } = await res.json();  
 window.WebChat.renderWebChat(  
 {  
 directLine: window.WebChat.createDirectLine({ token }),  
 userID: 'user1'  
 },  
 document.getElementById('webchat')  
 );  
 })();  
</script>

3. Upload the updated index.html file to your $web container in Azure Storage.  
4. Visit your static website URL to see the integrated chatbot.

## 7. Notes and Best Practices

- Do not expose your Direct Line Secret publicly in production.  
- For production, generate tokens on a secure server (Azure Function).  
- Use the same region for all services for better performance.  
- Free tiers may have usage limits.