

Azue Static Web App Bicep

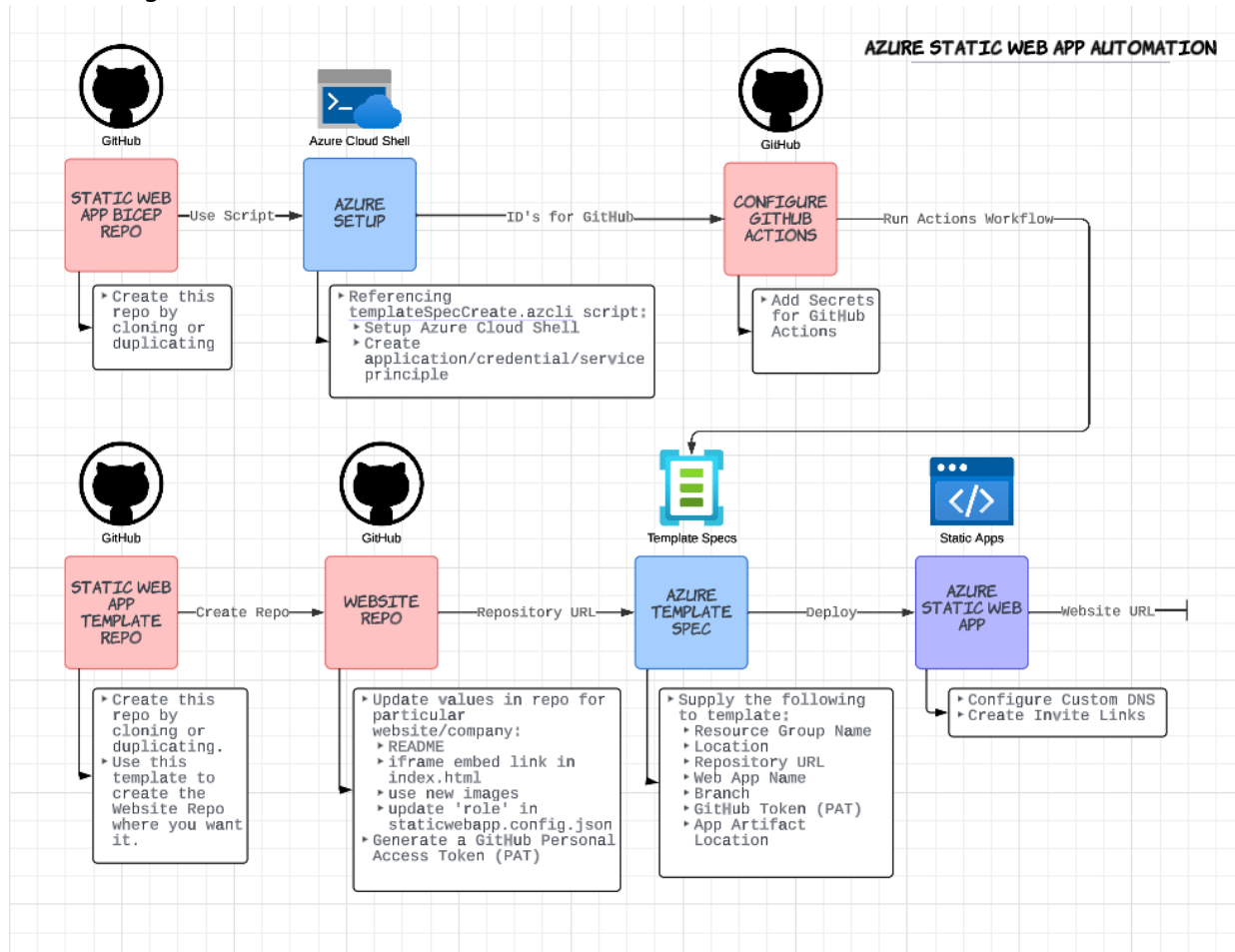
This repo contains the Bicep code for deploying an Azure Static Web App from your GitHub account/organization.

GitHub Actions are used to deploy the Azure Template Spec that is created from the Bicep code in this repo.

Once the Azure Template Spec is deployed, a Template Repository in GitHub is used to create as many website repos as you want. You will customize the site to make it unique. After setting up a GitHub PAT you can use the Azure Template Spec, along with your website information to deploy you custom Azure Static Web App. Once deployed you can set up custom DNS and invite users.

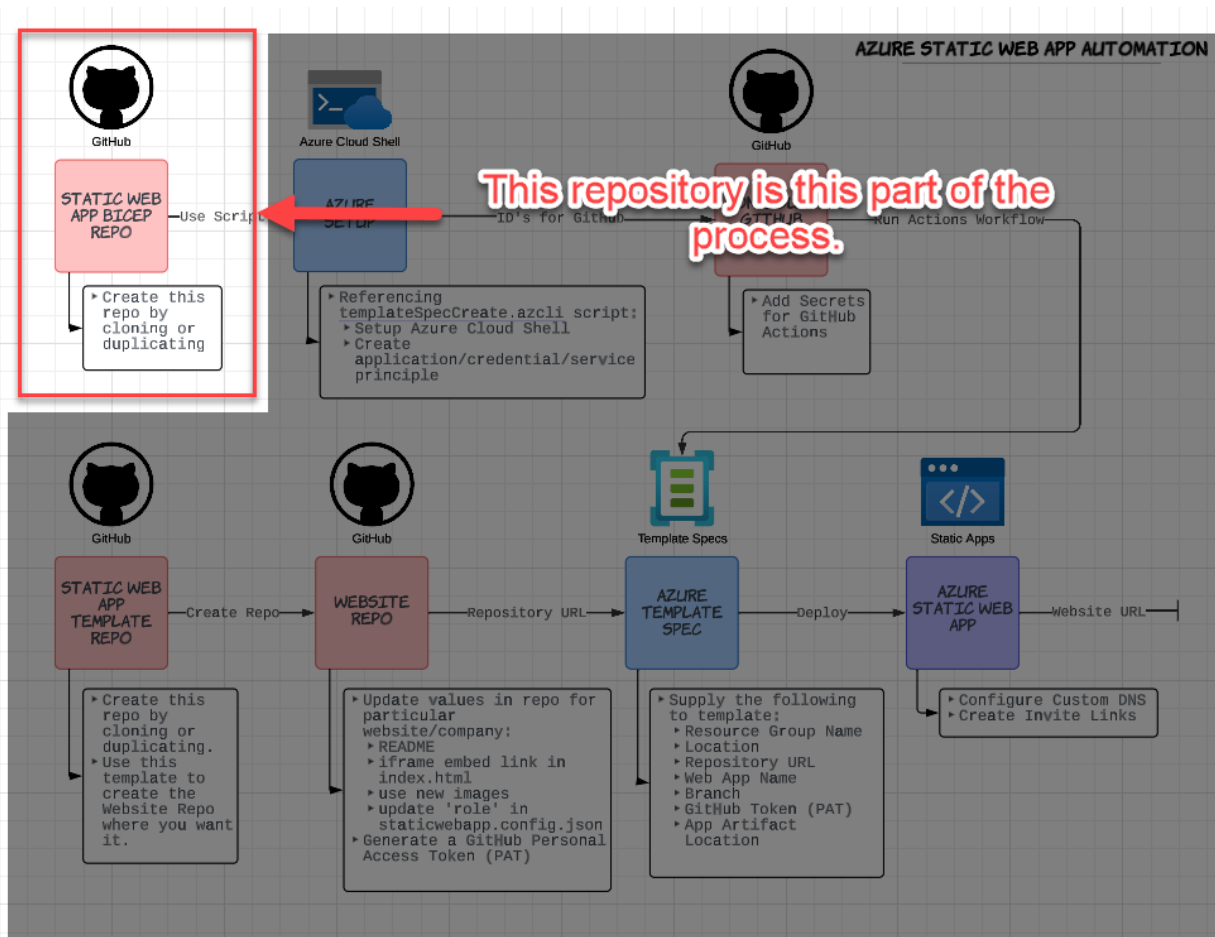
Documentation for this repo can be found in the [Documentation](#) folder.

Overview of the steps to setup Azure to deploy Azure Static Web Apps from your GitHub account/organization:



Step 1: Create a Copy of this repo

You need to copy this repo to your GitHub account/organization so that you can make changes to the Bicep code and GitHub Actions to deploy the Azure Template Spec.



Instructions to do this are in the **Documentation** folder, in the file **01 - How to Setup Static Web App Bicep Repo.md**.

1. Navigate to the GitHub repo you want to clone.
2. Click the green **Code** button.
3. Click **Download ZIP**.
4. Unzip the file.
5. Navigate to GitHub and create a new repo. Name it: **azure-static-web-app-bicep**
6. Navigate to the repo you just created.
7. Click the green **Code** button.
8. Click **Upload files**.
9. Drag and drop the files from the unzipped folder into the upload window.
10. Click **Commit changes**.

Step 2: Setup Azure to Deploy Azure Template Spec

You need to setup Azure to deploy the Azure Template Spec that is created from the Bicep code in this repo. This template spec will be used to deploy Azure Static Web Apps from your GitHub account/organization.


```
applicationRegistrationAppId=$(echo $applicationRegistrationDetails | jq -r '.appId')
```

5. Create the federated credential for the application registration:

```
az ad app federated-credential create \  
  --id $applicationRegistrationObjectId \  
  --parameters "{\"name\":\"azure-static-web-app-  
bicep\", \"issuer\":\"https://token.actions.githubusercontent.com\", \"subject\": \"repo:${githubOrganizationName}/${githubRepositoryName}:ref:refs/heads/main\", \"audiences\": [\"api://AzureADTokenExchange\"]}"
```

6. Create the service principal for the application registration:

```
az ad sp create --id $applicationRegistrationObjectId
```

7. Create the resource group that will be used to deploy the Azure Template Spec:

```
az group create --name 'rg-templateSpec' --location 'eastus2'  
resourceGroupResourceId=$(az group show --name rg-templateSpec --query id -  
-output tsv)
```

8. Create the role assignment for the application registration:

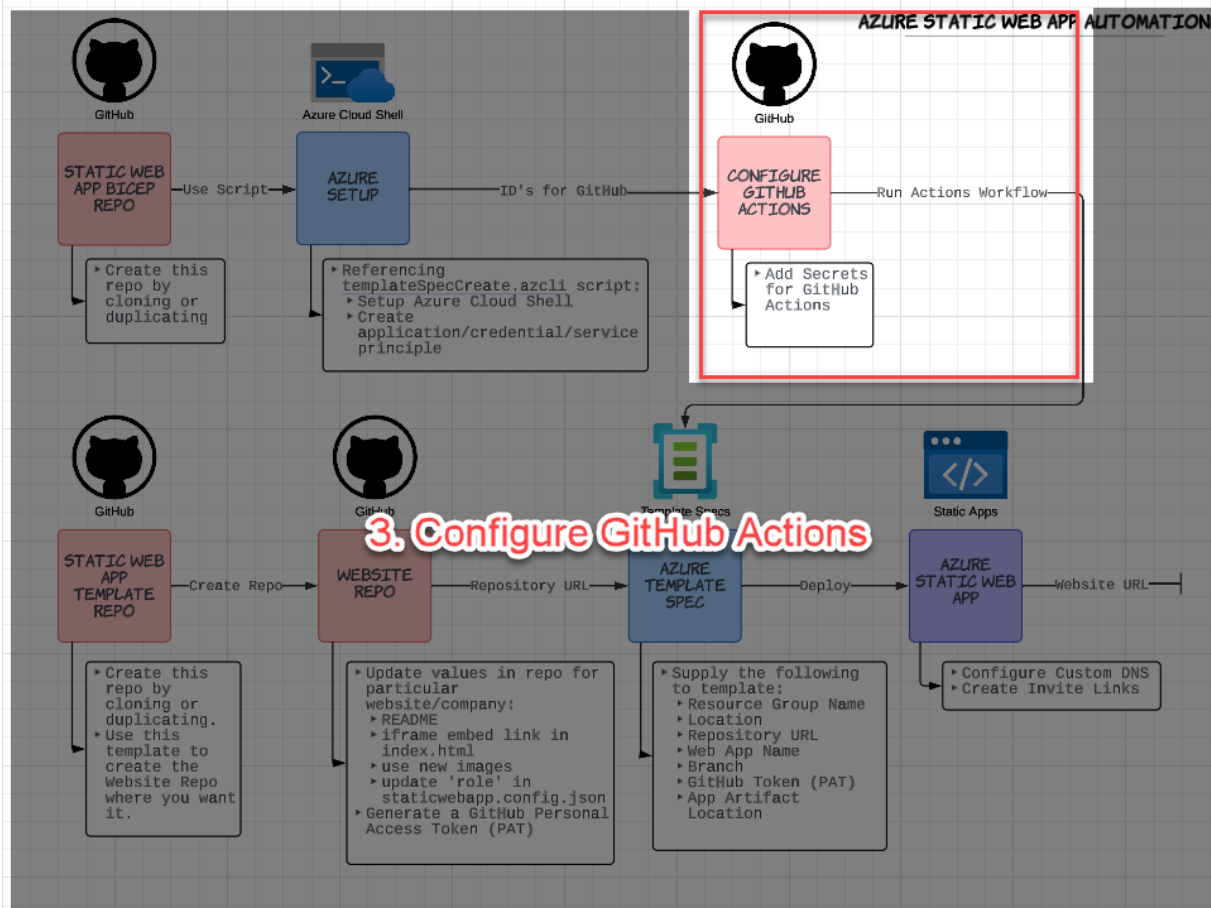
```
az role assignment create \  
  --assignee $applicationRegistrationAppId \  
  --role 'Contributor' \  
  --scope $resourceGroupResourceId
```

9. Echo the application registration details:

```
echo "AZURE_CLIENT_ID: $applicationRegistrationAppId"  
echo "AZURE_TENANT_ID: $(az account show --query tenantId --output tsv)"  
echo "AZURE_SUBSCRIPTION_ID: $(az account show --query id --output tsv)"
```

Step 3: Create Secrets for GitHub Actions & Deploy Template Spec

You need to create secrets for GitHub Actions that will be used in the Azure Template Spec to deploy Azure Static Web Apps from your GitHub account/organization. After the secrets are created, you will run the GitHub Action to deploy the Azure Template Spec.



Instructions to do this are in the **Documentation** folder, in the file **03 - How to Configure GitHub Actions to Deploy the Template Spec.md**.

1. Navigate to the GitHub repo **azure-static-web-app-bicep**.

2. Click **Settings**.

3. Click **Secrets and variables**.

4. Click **New repository secret**.

5. Click **Actions**.

6. Click **New repository secret**.

7. Create a secret for the Azure Client ID:

Name: **AZURE_CLIENT_ID**

Value: **The Client ID of the Application Registration you created in the previous step.**

8. Create a secret for the Azure Tenant ID:

Name: **AZURE_TENANT_ID**

Value: **The Tenant ID of the Application Registration you created in the previous step.**

9. Create a secret for the Azure Subscription ID:

Name: **AZURE_SUBSCRIPTION_ID**

Value: **The Subscription ID of the Azure Subscription you used previous step.**

10. Run the GitHub Action to deploy the Azure Template Spec:

Navigate to the GitHub repo **azure-static-web-app-bicep**.

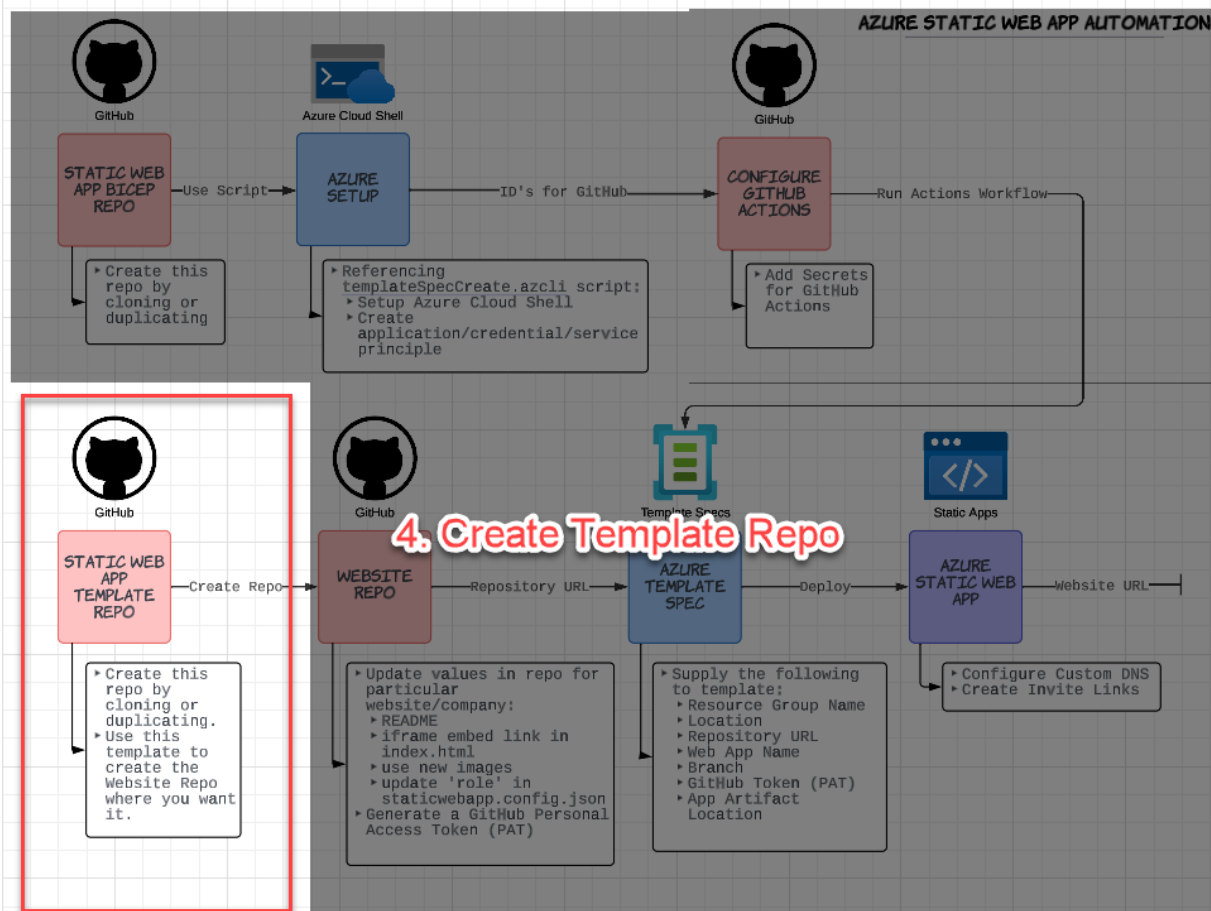
Click **Actions**.

Click **template-spec-static-web-app**.

Click **Run workflow**.

Step 4: Create an Azure Static Web App Template Repo

You need to create a repo that will be used to create an Azure Static Web App Template.

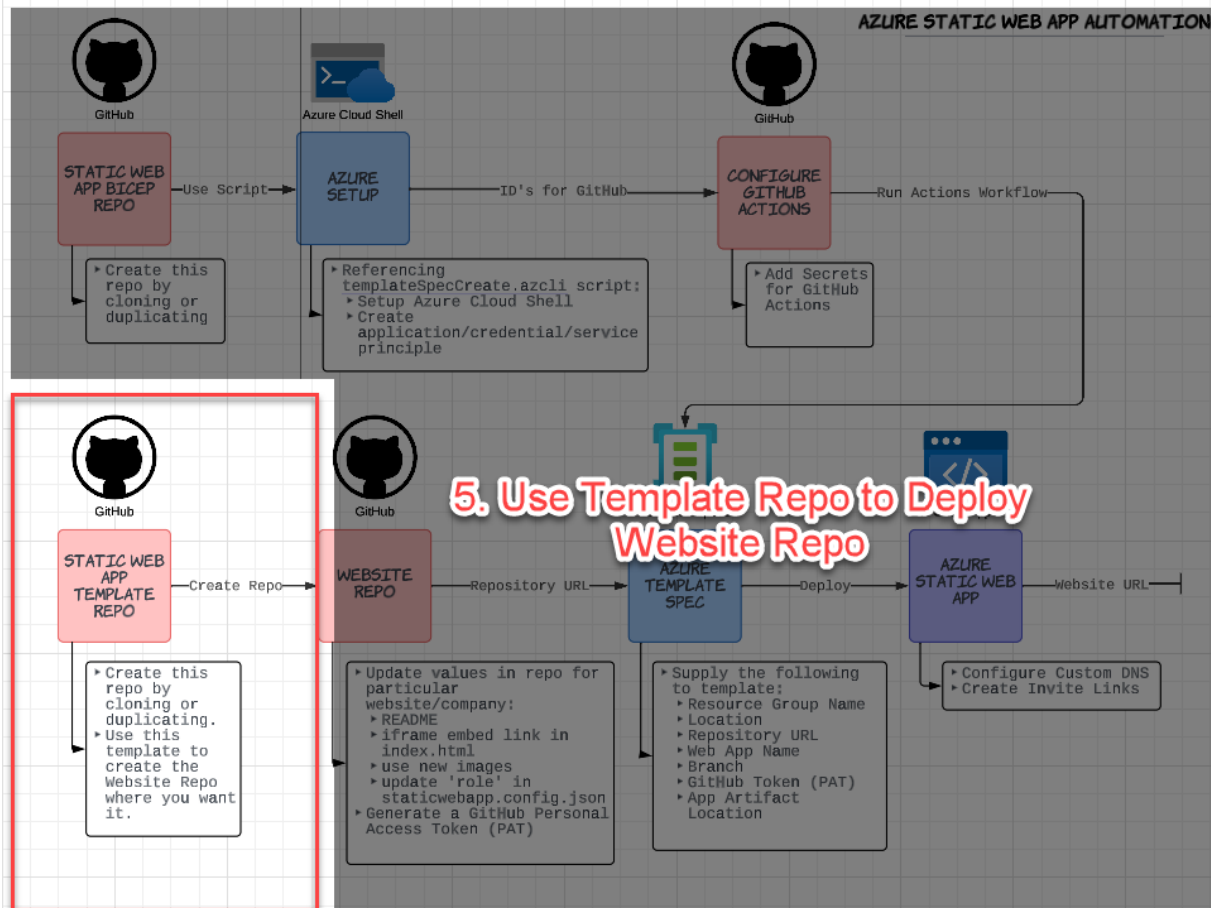


Instructions to do this are in the **Documentation** folder, in the file **04 - How to Create an Azure Static Web App Template Repo.md**.

1. Navigate to the GitHub repo you want to clone.
2. Click the green **Code** button.
3. Click **Download ZIP**.
4. Unzip the file.
5. Navigate to GitHub and create a new repo. Name it: **static-webapp-template**
6. Navigate to the repo you just created.
7. Click the green **Code** button.
8. Click **Upload files**.
9. Drag and drop the files from the unzipped folder into the upload window.
10. Click **Commit changes**.

Step 5: Use Template Repo to Create Website Repo

Using the template repo you just created, you will create a new repo that will be used to create an Azure Static Web App.

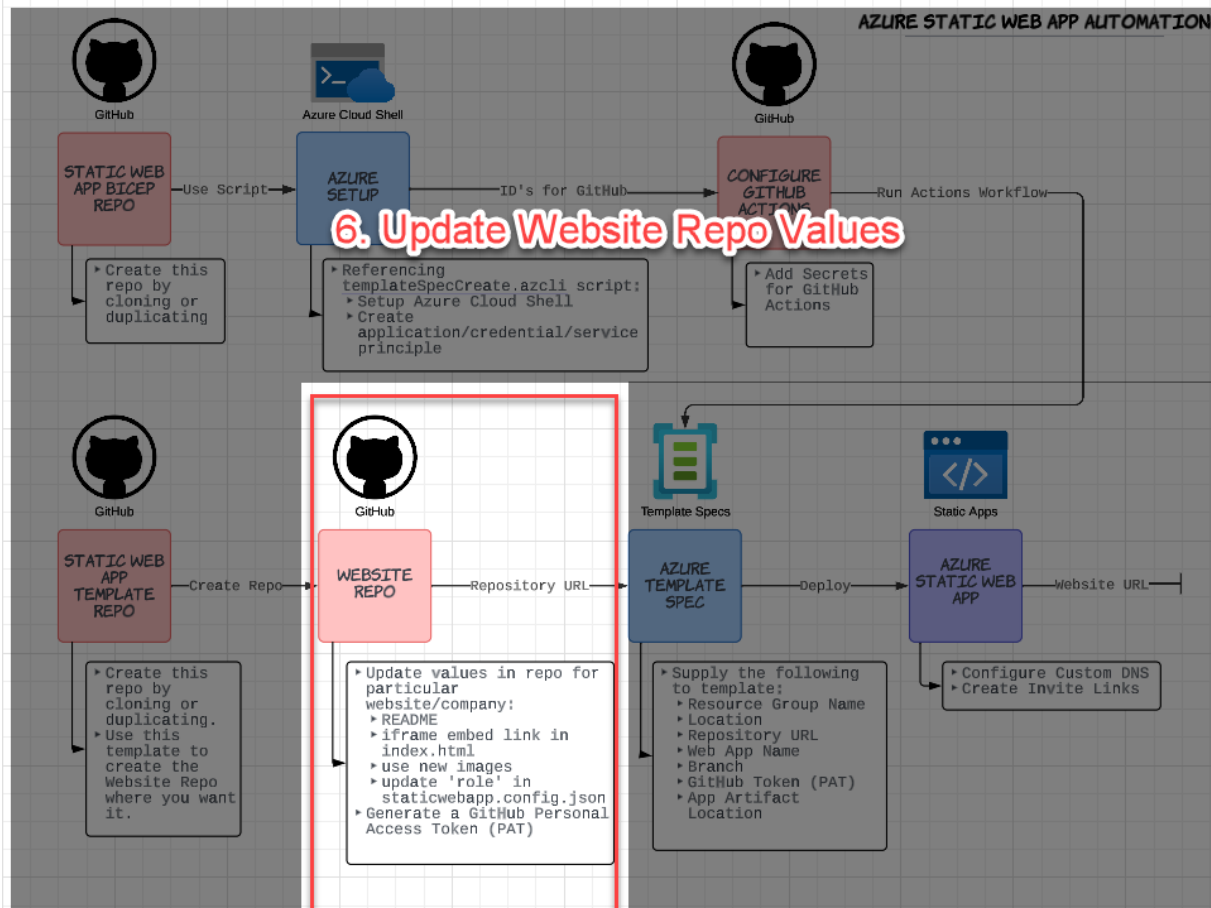


Instructions to do this are in the **Documentation** folder, in the file **05 - How to Use Template Repo to Create a Website Repo.md**.

1. Navigate to the template repo you created in the previous step.
2. Click the green **Use this template** button.
3. Click 'Create a new repository'.
4. Name the repo: **static-webapp-<Company Name>**.
5. Click **Create repository from template**.

Step 6: Update Values in the Website Repo

You need to update values in the newly created repo, like the name of the company, authentication role label, and the proper URL for the Power BI report. You also need to generate a GitHub Personal Access Token (PAT).



Instructions to do this are in the **Documentation** folder, in the file **06a - How to Update Values in the Website Repo.md**. Instructions for the PAT are in the file **06b - How to Create A GitHub PAT.md**.

Update Values in the Website Repo

1. Navigate to the website repo you created in the previous step.
2. Click on the README.md file, and click the pencil icon to edit the file.
3. Update the document title to match the company name.
4. Update the description as necessary.
5. Click **Commit changes** and then **Commit directly to the main branch**.
6. Click on the index.html file, and click the pencil icon to edit the file.

Make the following changes:

- Change the title on line 5 to match the company name.
- Change lines 35-36 to the appropriate values.
- Change the URL on line 40 to the appropriate publicly hosted Power BI Report.

7. Click **Commit changes** and then **Commit directly to the main branch**.

8. Click on the staticwebapp.config.json file, and click the pencil icon to edit the file.

Make the following changes:

- Change the **allowedRoles** value on line 5 to the appropriate value. This value will be used later to assign to users of your website.

9. Click **Commit changes** and then **Commit directly to the main branch**.

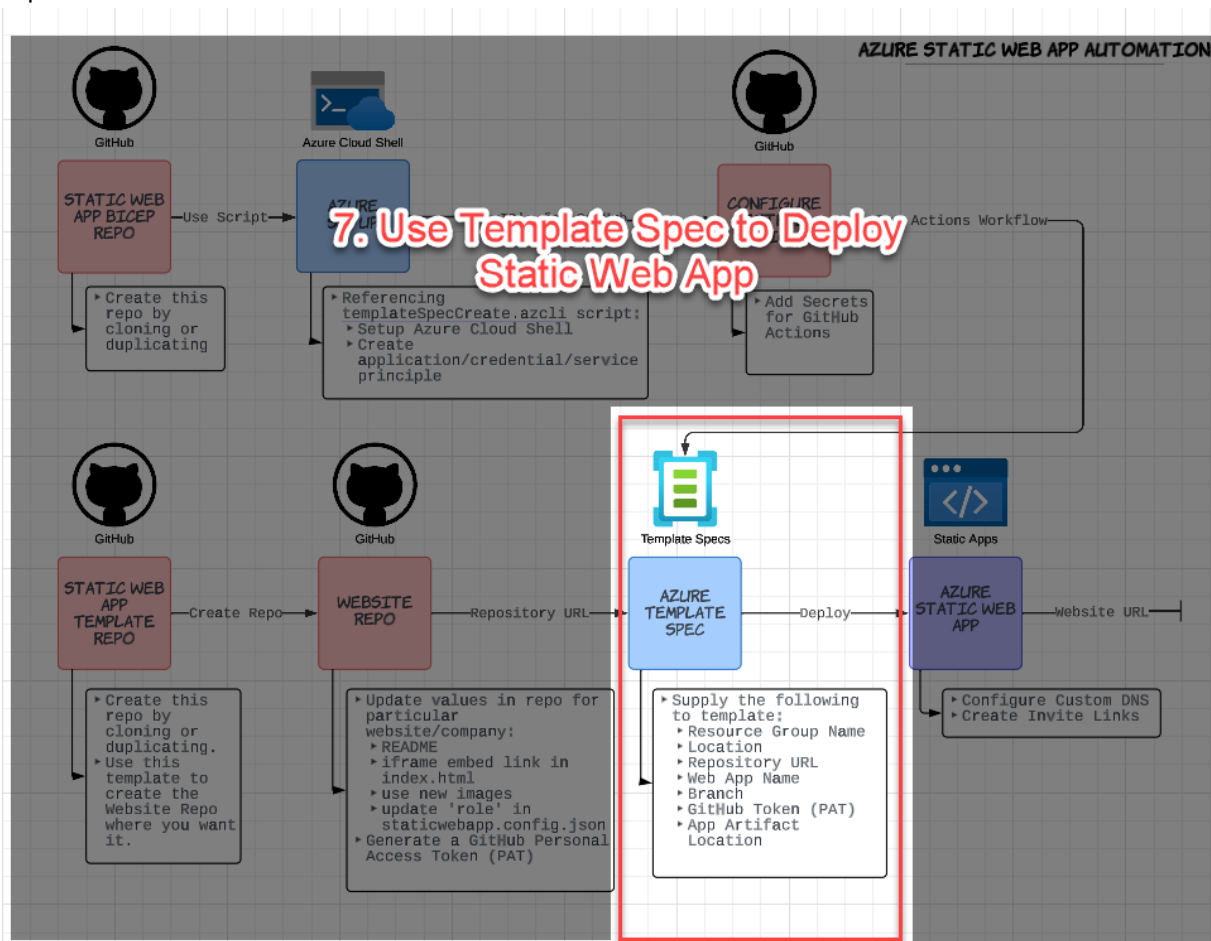
10. Replace the image in the **images** folder with the appropriate images. The FavIcon should be a 32x32 pixel image.

Create a GitHub Personal Access Token (PAT)

1. In the upper-right corner of any page, click your profile photo, then click **Settings**.
2. In the left sidebar, click <> **Developer settings**.
3. In the left sidebar, under Personal access tokens, click **Tokens (classic)**
4. Select Generate new token, then click **Generate new token (classic)**.
5. Give your token the name `static-web-app-deploy`.
6. To give your token an expiration, select the Expiration drop-down menu, then click a default or use the calendar picker.
7. Select the scopes you'd like to grant this token. For this use-case give it: `admin:repo_hook`, `repo` and `workflow`.
8. Click **Generate token**.
9. Copy the token to your clipboard. For security reasons, after you navigate off the page, you will not be able to see the token again.

Step 7: Create an Azure Static Web App Using the Template Spec

You need to create an Azure Static Web App using the Azure Template Spec that was deployed in the previous step. You will do this in the Azure Portal using information about your repo and the GitHub PAT that you created in the previous step. The Template spec will deploy your website, and automatically create the GitHub Action to deploy your website when you push changes to your repo.

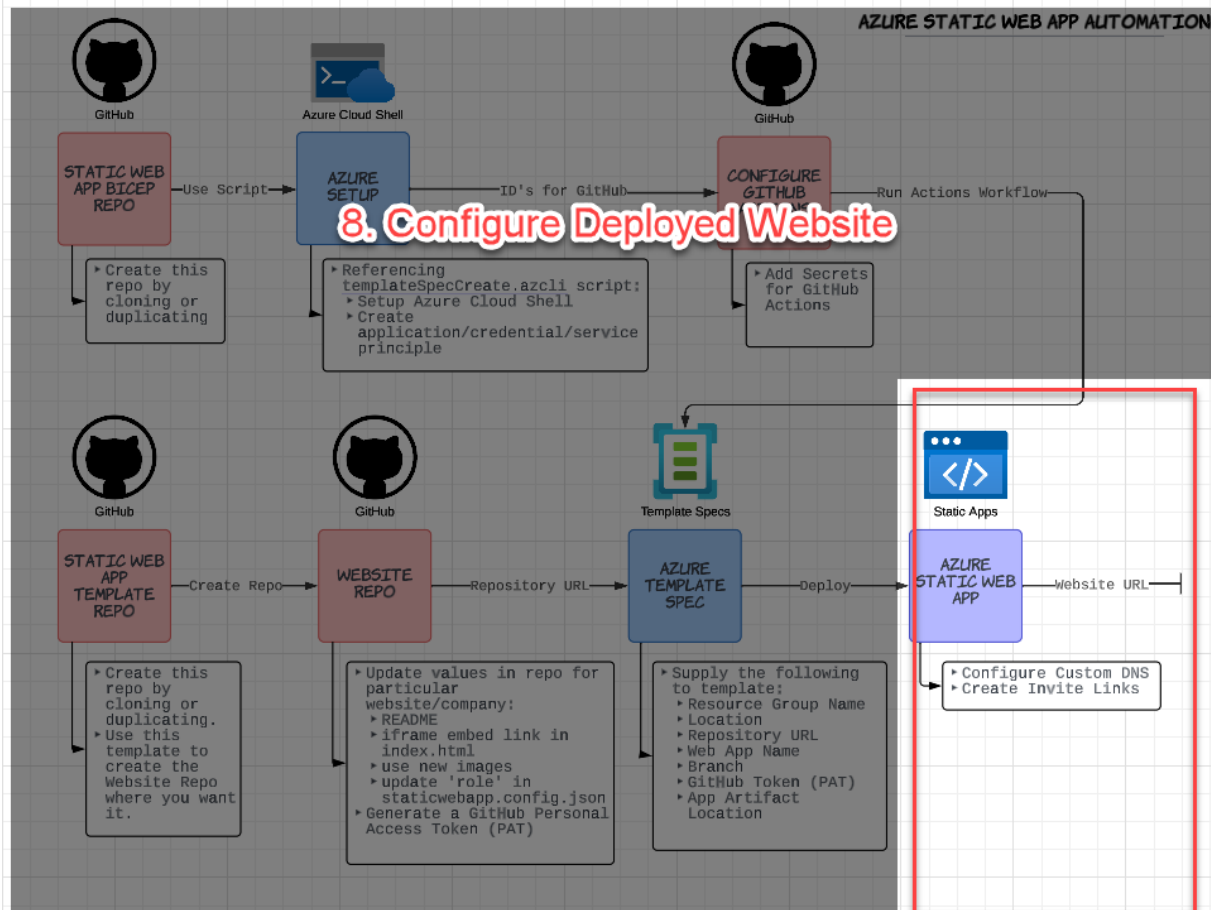


Instructions to do this are in the [Documentation](#) folder, in the file [07 - How to Create an Azure Static Web App Using the Template Spec.md](#).

1. Navigate to the Azure Portal.
2. Navigate to the Azure Template Spec. It will be in the `rg-templateSpec` resource group you created in a previous step, and it will be named: `static-web-app`. Click on the name of the Template Spec.
3. Click **Deploy**.
4. Update the following parameters:
Subscription: `<Your Azure Subscription>`
Region: `<Your Azure Region>`
Resource Group Name: `rg-<Company's Name>`
Location: `<The region you want the app to be deployed too >`
Git Hub Repo URL: `<The URL of the website repo you modified in a previous step>`
Web App Name: `static-web-app-<Company's Name>`
Git Hub Branch: `main`
Git Hub Token: `<The GitHub Token you created in a previous step>`
App Artifact Location: `\`
5. Click **Review + Create**.
6. Click **Create**.
7. Click **Go to resource**.

Step 8: Configure the Azure Static Web App

You need to configure custom DNS for the Azure Static Web App, and then configure the Azure Static Web App to use the custom DNS. You will then need to create invite links for the users that will be able to access the Azure Static Web App.



Instructions to do this are in the **Documentation** folder, in the file **08 - How to Configure the Azure Static Web App.md**.

1. Navigate to the Azure Portal.
2. Navigate to the Azure Static Web App you created in the previous step. It will be in the **rg-
<Company's Name>** resource group you created in a previous step, and it will be named: **static-
web-app-<Company's Name>**. Click on the name of the Static Web App.
3. Click **Custom domains**.
4. Click **Add** and choose **Custom domain on other DNS**.
5. Enter the domain you want to use for the Static Web App.
6. Click **Next**.
7. Click **Generate Code** to generate a DNS TXT record. Generate and copy the TXT hostname record and enter it with your DNS provider to confirm your domain ownership. It can take up to 12 hours for DNS entry changes to take effect.
8. Once the TXT record has propagated, you will repeat the above steps but this time you will change the **Hostname record type** to **CNAME**.
9. Click **Add**