



First Look: Azure Static Web Apps

VAIBHAV GUJRAL
MICROSOFT MVP - AZURE

About Me



- Cloud Architect at Kiewit Corp
- Microsoft Most Valuable Professional (MVP) – Azure
- Microsoft Certified Trainer (MCT)
- Microsoft Certified Azure Solutions Architect Expert
- Organizer, Omaha Azure User Group
- <http://www.vaibhavgujral.com>
-  [@vabgujral](https://twitter.com/vabgujral)
-  [linkedin.com/in/vaibhavgujral/](https://www.linkedin.com/in/vaibhavgujral/)
- #AzureHeroes Community & Content Hero
- Listed among the “Top 50 Microsoft Azure Blogs, Websites & Influencers in 2020”



Agenda

- Just First impression of Azure Static Web Apps

What are Azure Static Web Apps?

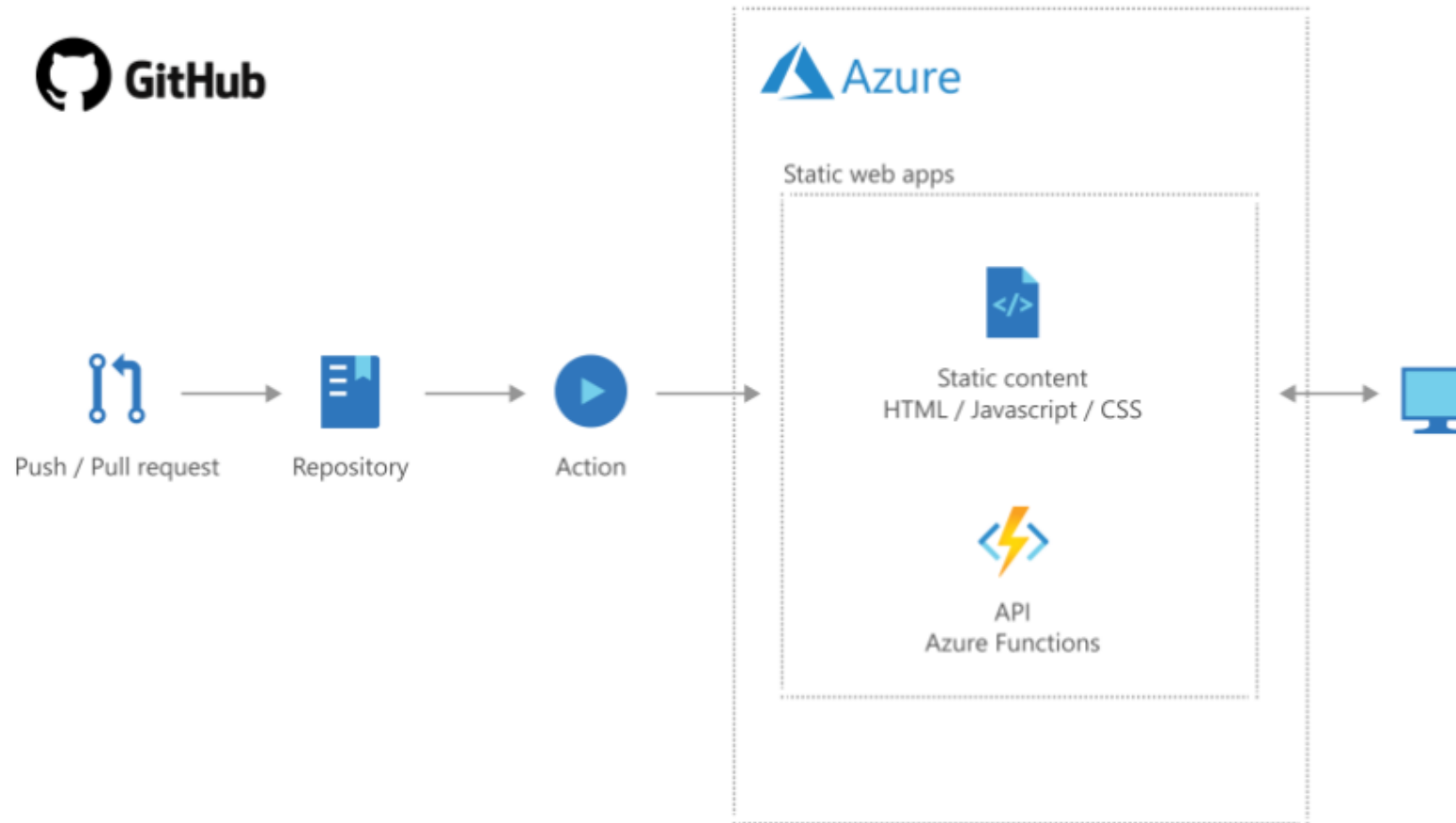


Azure Static Web Apps is an Azure service that automatically builds and deploys full stack web apps to Azure from a GitHub repository.

Features:

1. Ideal for static web apps built using libraries and frameworks like Angular, React, Vue or Blazor.
2. Two pieces to it
 1. Static Content – which is served through geographically distributed locations
 2. API endpoints – using Azure functions
3. Seamless GitHub Integration (yes, no support for Azure DevOps as of now)
4. Supports static site generator – Gatsby, Hugo, VuePress

What are Azure Static Web Apps?



Routes



Defined as an array of rules in *routes.json* file located at the root of the build artifact directory

Routes in static web apps lets you define the following two things:

1. Back-end routing rules
2. Authorization roles

Rules are executed in the sequence as they appear in the array

Roles are defined in *routes.json* and users are associated to roles through invitations

Restrict Access



Use Invitations to associate users with roles and authorize users to routes based on rules defined in *routes.json*

Following providers are supported:

1. Azure Active Directory
2. GitHub
3. Facebook
4. Google
5. Twitter

Two built-in roles:

1. anonymous
2. authenticated

API Support

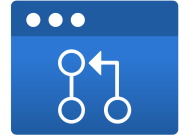


Azure Static Web Apps provides serverless API endpoints via Azure Functions

Limitations –

1. The API route prefix must be api
2. API must be a javascript, C#, or Python Azure Function App
3. Route rules for API functions only support redirects and securing routes with roles
4. Only HTTP triggers are supported
5. Logs are only available with Application Insights

Environments



Pre-production environments in ASWA provides a full-fledged staging environment

When a pull request is created against a branch that the Actions workflow watches, a pre-production environment is built

Once you merge your pull request, the changes are merged to the production environment

Multiple pre-production environments can co-exist at a time*

Staged versions of your application are currently accessible publicly by their URL, even if your GitHub repository is private.

* - During preview, only one staging environment is allowed

Demo



Thoughts



1. Very easy option to deploy your static apps from Github
 - Automatically sets up GitHub Action
2. Supports many popular frameworks
3. Integration with Azure Functions - 100
4. Still in preview – so not recommended for production
5. Only supports free tier currently
6. Environments looks promising

Resources / references



1. Azure Static Web Apps Documentation: <https://docs.microsoft.com/en-us/azure/static-web-apps/>
2. GitHub Actions: <https://docs.github.com/en/actions>
3. Blazor: <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>
4. Blazor and C# APIs - <https://docs.microsoft.com/en-us/azure/static-web-apps/deploy-blazor>
5. GitHub Repo: <https://github.com/vabgujral/ASWA-Blazor-demo>

thank
you