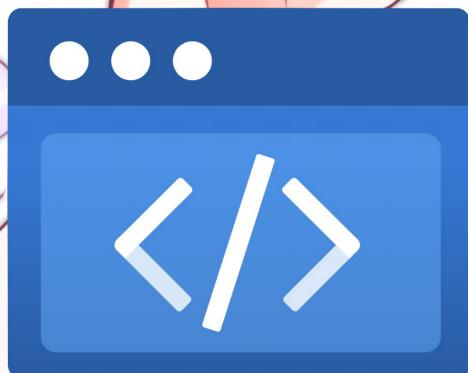


マンガでわかる

# Azure Static Web Apps



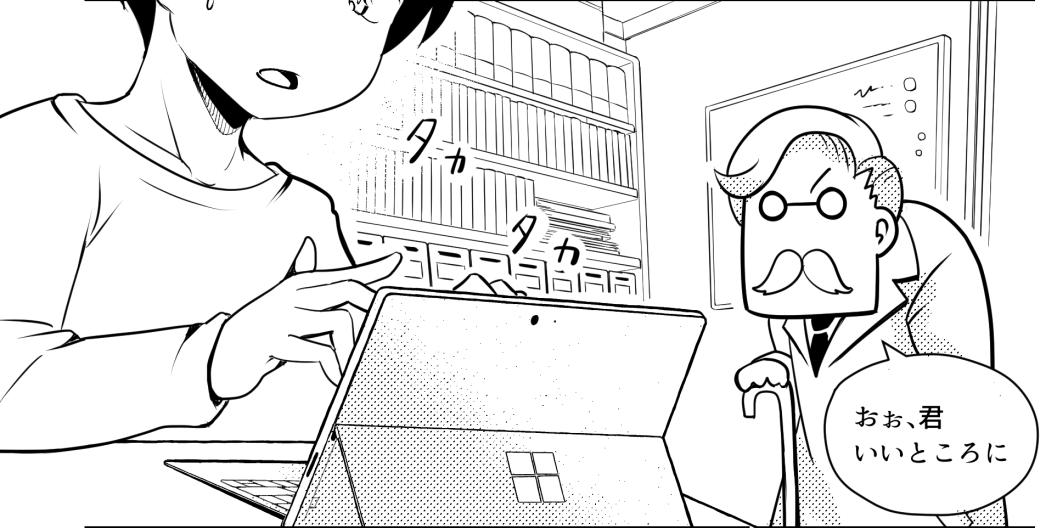
クラウド  
初心者でも  
できる⑨



CIツール不要の自動デプロイ

**Go to next page**





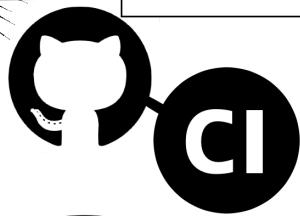
教授の無茶振り、  
勢いで  
引き受けちゃった  
けど……

クラウドって  
必要な知識が  
多すぎて…



しかも  
自動デプロイって  
どうやるの！？

とりあえず  
検索してみよ



# シー・アイ ギットハブ CIツールと GitHub を トークンで繋ぐ…？

Config.yml

そこで、CIに命令するための  
yamlっていうファイルを  
書く必要がある…？

まったく  
わからぬじ  
どうしよ…



それなら

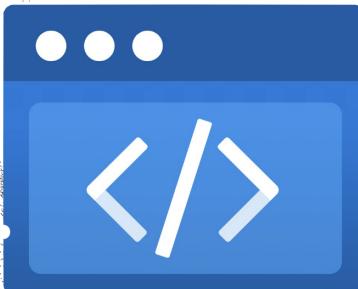
アジュール スタティック ウェブ アップス  
**Azure Static Web Apps !!**

わあ、ソフィア！  
なんで僕の大学に…？

っていうか  
あじゅーる  
すたていっく…?  
って何？

Microsoft の  
新サービスだよ

必要なのは GitHub と  
キットハブ  
アジュール  
Azure のアカウントだけ



GitHub にコードが  
上がっていれば  
そのまま使えるよ！

ええっと、  
トークンで繋いだり  
yamlを書いたり  
しなくても  
いいの……？

そう！  
リポジトリを選んで  
クリックしていくだけ

1分あれば  
公開できるんだよ

my-first-  
static-web-app

1分？ 本当に～？  
僕、ちゃんと  
ストップウォッチで  
計ってるからね

まかせて！

ほら見て！

公式ドキュメントに  
サンプルリポジトリが  
用意されているから  
コードを書かなくても  
いきなり試せるんだ

## リポジトリを作成する

この記事では、GitHub テンプレート リポジトリは、さまざまなフロントエンド フレームワーク

Angular React Vue フレームワーク

- GitHub にログインしていることを確認  
<https://github.com/staticwebdev/angular>
- リポジトリの名前を `my-first-static-w`

フレームワークを  
アンギュラー、React、  
ピュージェイエス、  
Vue.js から選べるけど  
どれがいい？



<https://aka.ms/first-static-web-app>

Vueなら  
さわったことが  
ある

Vue に  
するよ

ホチ

リポジトリが  
できた！

ここまで10秒！

Vue B

ア ジ ュ ー ル ポ ー タ ル  
次は Azure Portal に移動して  
このリポジトリを登録するよ

<https://portal.azure.com/>

Microsoft Azure

リソース、サービス、ドキュメントの検索

Azure サービス



リソースの作成



すべてのリソース



Virtual Machines



App Services

[リソースの作成]をクリックしたら  
**Static** って検索してみて

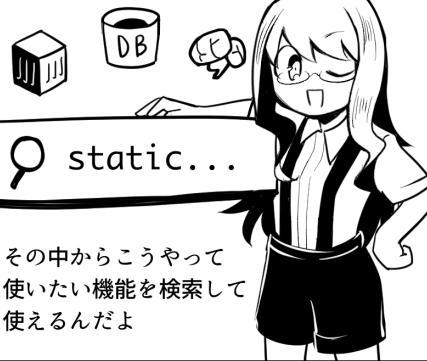
お！

出てきた出てきた

これをクリックして  
[作成]だね

自分のサブスクリプションと  
リソースグループ、アプリの名前、  
最後にリージョン(サーバーの場所)を選んだら

ア ジ ュ ー ル  
Azure ってひとくちに言っても  
ストレージやコンテナ、データベース  
人工知能までたくさんのサービスが  
提供されているんだけど



結果をすべて表示

静的 Web アプリの詳細

名前 \* my-first-static-web-app

地域 \* East Asia

SKU \* Free

GitHub アカウントでサインイン

サブスクリプション \* 従量課金

リソース グループ \* (新規) seminar\_website 新規作成

静的 Web アプリの詳細

名前 \* my-first-static-web-app

地域 \* East Asia

SKU \* Free

GitHub アカウントでサインイン

[GitHubアカウントでサインイン]!!

なるほど！



AzureとGitHubが  
ここで直接つながるから  
トークンの登録作業は  
必要ないんだ！

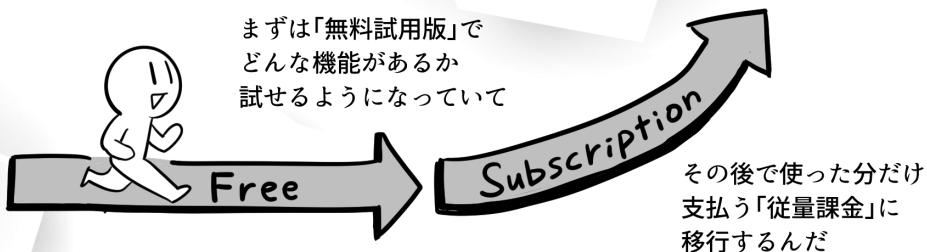
かんたん♪

ところで  
サブスクリプション  
ってなんだろう



聞いたこと  
ないなあ

サブスクリプションは  
クラウドの契約形態のこと



\* Static Web Apps は、プレビュー期間中は無料プランのみ

リソースグループについても最初は  
よくわからないから  
説明が必要だね

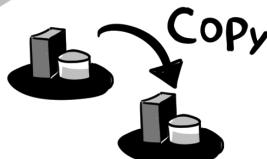
Resource group

- ・サーバー
- ・ストレージ
- ・ネットワーク

といった  
コンピューターの  
「リソース」を  
まとめて管理  
できる仕組みだよ

リソースグループは  
テンプレート化して  
コピーしたり

使わなくなったら  
一括で削除したりできるから  
便利だよ



自分でわかりやすい名前を  
つけられるんだね！  
僕は seminar\_website と  
してみたよ

さて、GitHubを連携したから  
リポジトリが選べるようになつたよ

llminatoll

my-first-static-web-app

master

次: ビルド >

さつき作った  
リポジトリを選んで…  
公開プランチは  
masterプランチでいいよね

最後にビルドの設定を  
ちょこっと  
指定したら完了！

ここまで40秒！

基本 ビルド タグ 確認および作成

初期ビルドの変数を指定します。これらは、後からワークフロー ファイルで変更できます。

ビルドの詳細

アプリの場所 \* ①

API の場所 ①

アプリの成果物の場所 ①

確認および作成

/

例: "api"、"functions" など…

dist

ここはフレームワークによって  
少しずつ違うんだけど  
全部公式ドキュメントに  
書き方が載ってるからね

…こ、  
これは!?

いきなりビルドが  
始まつたん  
ですけど!?

「作成」ボタンを  
押したらすぐに!!

yamlも書いて  
ないのに!!

びっくりした~

そう言ってる間に  
ビルドが完了したよ

はや!!

はやく公開された  
Webサイトを見たいよね

✓ デプロイが完了しました

デプロイ名: Microsoft.Web-StaticApp-Portal-68b...  
サブスクリプション: 従量課金  
リソース グループ: seminar\_website

▽ 展開の詳細 ([ダウンロード](#))

△ 次の手順

[リソースに移動](#)

ここをクリックして  
表示されたURLへ  
行ってみよう!

わあ!  
本当にもう  
公開されてる!



ちょうど1分だ  
.....

完璧だよ  
ソフィア

ちなみにビルドの状態は  
GitHub上で確認  
できるからね



ちなみにビルドの状態は GitHub上で確認できるからね

Azure Static Web Apps CI/CD

This run Workflow file

2 completed jobs in 1m 56s

Artifacts

わざわざCIツールを開かなくてもいいとは!

<https://black-ocean-xxx.azurestaticapps.net/>

あ！見て！  
自動生成されたURLが  
black-ocean だって！  
かっこいい！

これ以外も  
バリエーションが  
あるのかなあ？

他にはこんな  
パターンがあるよ

polite-dune

mango-sky

happy-stone

ambitious-meadow

jolly-coast

gray-smoke

おもしろい！  
mango-skyなんて  
可愛いね！

でも、ランダムURLのままじゃ  
ちょっとなあ



もちろんカスタムドメイン\*も  
設定できるよ！  
証明書も無料！

本当!?  
よかった～

設定

構成

カスタムドメイン

関数

\* <https://aka.ms/custom-domain>

みっくん、  
盛り上がってるとこ悪いんだけど  
さっそく新規ページを  
プルリクしたから  
チェックしてくれる？



お、おう

えーと…  
更新分を見るには

プルリクされたブランチを  
プルしてきて  
自分のパソコン上で  
ビルドしてチェックする  
必要が…

その必要は  
ないよ！

## プルリクエストレビュー機能に おまかせ！

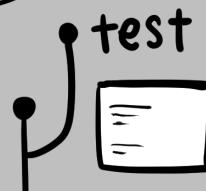
わざわざ手元で  
ビルドしなくとも  
非公開のURLで



プルリクが  
テストデプロイされたものを  
直接見られるよ★

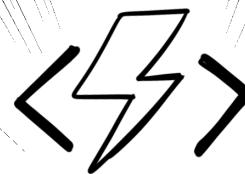
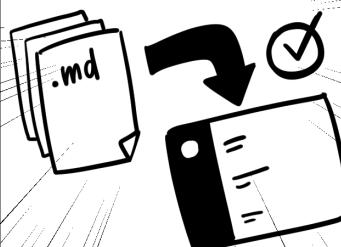


これは嬉しい機能だね！  
本来ならテスト環境を  
作るのにもひと苦労なんだよね



実は他にも  
Azure Static Web Appsには  
イケてる機能があるんだ  
たとえば…

ギャツビーヒュゴ  
Gatsby や Hugo のような  
スタティックサイト  
ジェネレーターでも  
自動でビルドが走る!!



Azure Functions



バックエンドの  
API を統合して  
一緒にデプロイできる!!

Facebook や  
Google アカウントを  
使った認証を  
組み込める!!



すごい  
自由度高い！

実は最初、『それなら GitHub Pages でいいんじゃない？』と思ったけど、GitHub Pagesには APIを統合する機能も認証機能もないよね

そうだね！  
あと、ドキュメントやチュートリアルが  
**すべて日本語化されているのも**  
他のツールと違うポイントだね

次の日

君たち、ゼミの  
Webサイトの  
進み具合は  
どうだね？

って、もう完成  
してるだと~!?

しかも  
Vue.jsで  
イケてる構成！  
テスト環境も  
あるなんて！

えへへ

実は、ほとんど  
Azure Static Web Apps の  
おかげなんだだけね★

★あなたも Azure Static Web Apps で、クラウドに爆速デプロイしよう！

# クラウドを始めたい人におすすめのチュートリアル

どこから始めるか迷っているあなたへ

## プログラミングの基礎



<https://aka.ms/Intro-to-Dev>

### Point

このラーニングパスのように  
好きなモジュールを組み合わせて  
あなただけの  
コレクションを作れるよ！

## Gitの概要



<https://aka.ms/Intro-to-Git>



## クラウドの基礎



<https://aka.ms/AFUN-LP>

## Web開発の基礎



<https://aka.ms/Deploy-a-Website>



## Azure Static Web Apps



<https://aka.ms/StaticWeb-API>



## Gatsbyで 静的Webアプリ



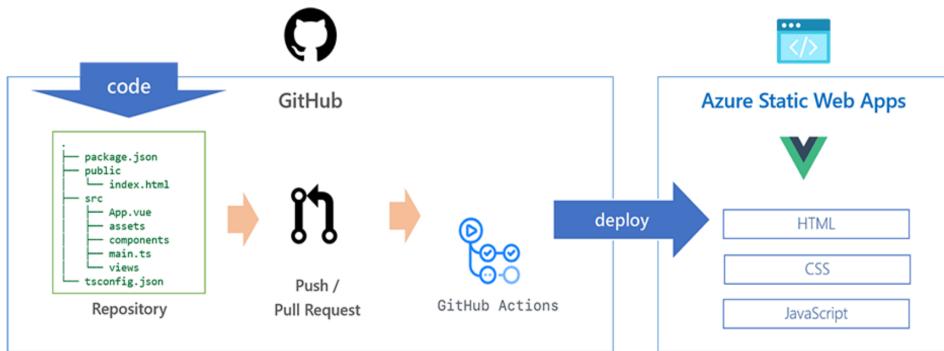
<https://aka.ms/StaticWeb-Gatsby>



漫画：湊川あい

# Vue.js で作るサイトを Azure Static Web Apps に CI/CD も含めて数分でホスティングする方法

Microsoft Build 2020 で Public Preview になったスタティックサイト向けの新しいホスティングサービスである Azure Static Web Apps を使うと Vue.js のアプリケーションを数分でデプロイしてクラウド上に公開できるようになります。



Vue.js はプログレッシブフレームワークというコンセプトで作られているため、最初は小さく始めて徐々に本格的な構成のアプリケーションに育てていくことができるようになっています。この記事では、Vue.js と Azure Static Web Apps を使ってスケーラブルなアプリケーションを作るための最初の一歩を紹介したいと思います。

## Vue CLI を使ったアプリケーションの新規作成

Azure Static Web Apps では、アプリケーションのコードが GitHub に存在していることが前提です。なので、まず基本的なコードを書いてしまいましょう。

Vue.js アプリケーションを新規に作るには Vue CLI を使うのが近道です。Vue CLI を使えば、例えば Router の導入や TypeScript の開発などを適切な方法で開始することができます。

では、早速 Vue CLI を実行してみましょう。ターミナルを使ってアプリケーションを作成するディレクトリに移動して以下のコマンドを実行します。

```
$ npm i -g @vue/cli  
$ vue create .
```

\$ vue create .  
Vue.js のプロジェクトを作成するよ

Vue CLI のプロンプトには、以下のように良く使う設定でプロジェクトを新規作成することにします。TypeScript は Vue の開発でも今後主流になると思うので選択しておきます。また Router や ESLint + Prettier、そして Jest も多くのプロジェクトで使われているので Vue CLI で最初から導入しておきます。

なお、TypeScript の **Use class-style component syntax?** は、Vue 3.x ではメンストリームから外れたので、ここでは No を選択したほうが良いでしょう。

```
Vue CLI v4.3.1
```

```
? Please pick a preset: Manually select features
? Check the features needed for your project: TS, Router, Linter, Unit
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? No
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Pick a linter / formatter config: Prettier
? Pick additional lint features: Lint on save
? Pick a unit testing solution: Jest
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
? Save this as a preset for future projects? (y/N) N
```

しばらくすると `Vue.js` のプロジェクトが生成されます。まずローカルでアプリケーションが起動することを確認します。

```
$ npm run serve
```

無事に `Vue.js` の緑のアイコンとともに初期画面が表示されれば、`Vue.js` のアプリケーション基本構成はできあがりです。TypeScript のコンパイルはもちろん、ユニットテストやフォーマッティングも既に実行できるようになっています。あとはプロダクト固有の機能を追加するコードを書くだけです。

この時点で GitHub にコミットして Azure にデプロイする準備をしておきましょう



## Azure Static Web Apps のプロビジョニング

Azure Static Web Apps は Azure Portal の Create a resource で static web apps と入力し、以下のように必要事項を入力します。この時、さきほど作成した GitHub リポジトリの情報をリンクさせます。

Static Web App (Preview) >

Basics Build Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered files are served from a global footprint with no web servers required. [Learn more](#)

**Project Details**

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ [Subscription dropdown]

Resource Group \* ⓘ [Resource Group dropdown: myk-mmld] [Create new](#)

**Static Web App details**

Name \* swa-vue

Region \* West US 2

SKU \* Free

**Source Control Details**

GitHub account k-miyake

Organization \* k-miyake

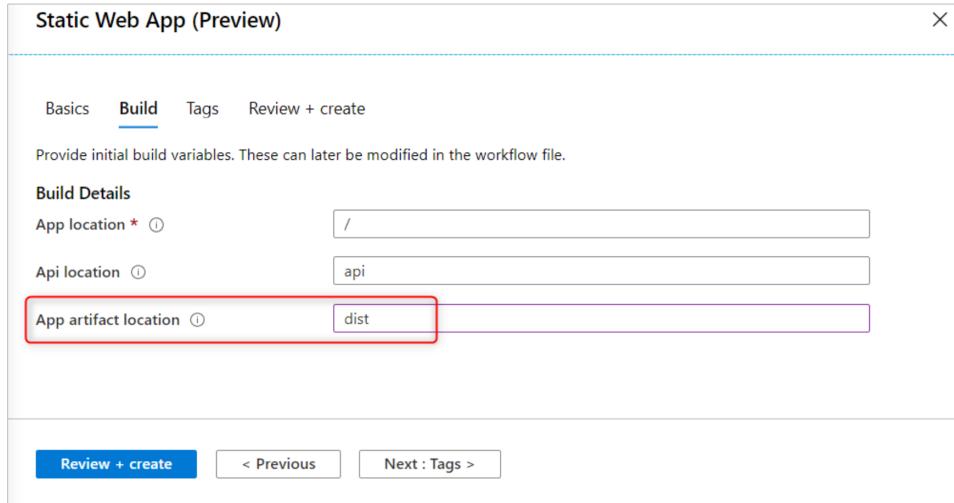
Repository \* azure-swa-vue

Branch \* master

[Review + create](#) < Previous Next : Build >



次の Build 設定画面では **App artifact location** に **dist** と入力します。dist は、Vue CLI 標準のビルトコマンドである `npm run build` を実行した際に HTML, CSS, JavaScript などホスティングの対象となるファイルが出力されるディレクトリです。



Static Web App (Preview)

Basics Build Tags Review + create

Provide initial build variables. These can later be modified in the workflow file.

Build Details

App location \* /

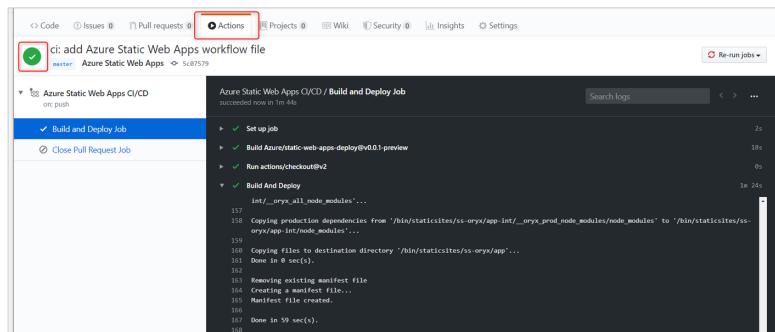
Api location api

App artifact location dist

Review + create < Previous Next : Tags >

あとは、**Review + create** をクリックして数分待てば、デプロイが完了です。デプロイのために CI サービスを設定したり、YAML でビルトスクリプトを書いたりする必要はありません！

なお、ビルトとデプロイはバックグラウンドで GitHub Actions のビルトワークフローが自動で実行されます。ビルトの状況を見るには、リポジトリの Actions タブを開いて確認できます。アイコンがグリーンになっていればビルトとデプロイが成功しています。

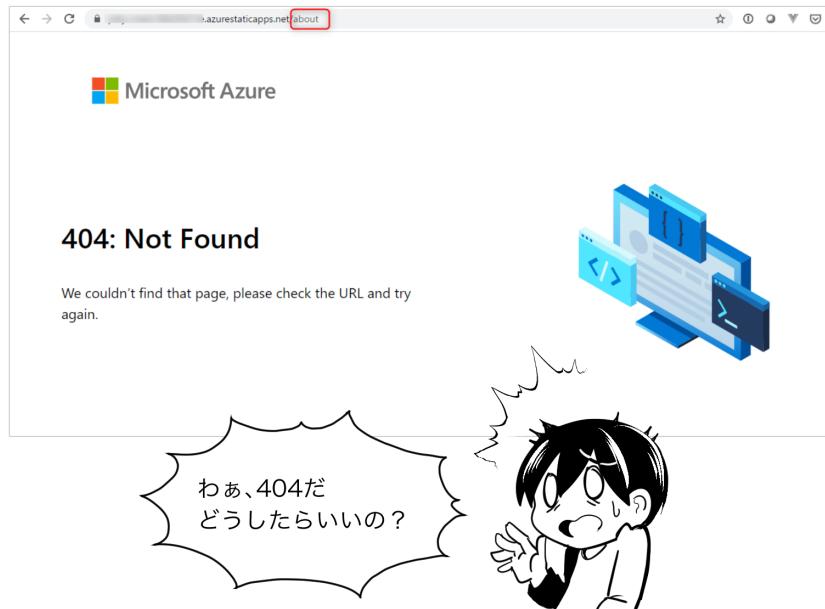


Azure Portal に戻って サイトの URL をクリックしてみましょう。ローカルで実行したアプリケーションと同じページが公開されているのを見ることができるのはずです。

この時点での GitHub リポジトリの `.github/workflows` に YAML ファイルが追加されているので、ローカルに Pull するのを忘れないようにしましょう。

## フォールバックを設定する

Vue CLI で Router を導入した際に、**History Mode** を選択しました。SPA ではサー  
バー側に何も設定をしなければ、特定のページ URL を直接指定すると 404 が発生してし  
まいります。



History Mode をホスティング環境でも有効にするには、その環境に適した設定を行う必  
要があります。Azure Static Web Apps では以下のような **routes.json** ファイルを  
**/public** 以下に作成します。



では、この変更をブランチにプッシュしてプルリクエストを作成しましょう。

プルリクエストをトリガーに Static Web Apps へのデプロイが実行されています。

## Enable fallback route #2

[Open](#)

k-miyake wants to merge 1 commit into `master` from `fallback-route`

Conversation 0

Commits 1

Checks 0

Files changed 1



k-miyake commented now

Owner ...

Add routes.json to enable the fallback

add routes file

3f4a7ff

Add more commits by pushing to the `fallback-route` branch on [k-miyake/azure-swa-vue](#).



Some checks haven't completed yet

1 queued check

[Hide all checks](#)

- Azure Static Web Apps CI/CD / Close Pull Request Job (pull\_request) [Queue...](#) [Details](#)

This branch has no conflicts with the base branch

Merging can be performed automatically.

[Merge pull request](#)

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

ビルドプロセスが完了すると、Azure ポータルの **Environments** から変更がデプロイされたアプリケーションを実際に確認することができるようになります。

Settings

- Configuration
- Custom domains
- Functions
- Environments**
- Role management
- Locks
- Export template

Production

There is one production environment per app. Check its status and last update time here.

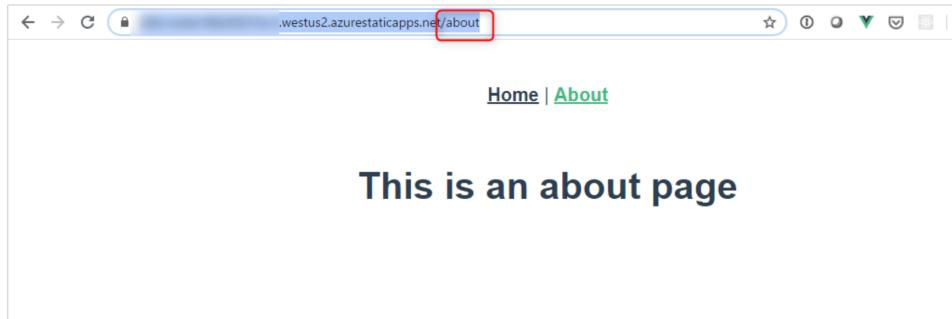
Branch	Last Update Time	Status	Browse
master	2020年5月25日 20:07:25 JST	Ready	<a href="#">Browse</a>

Staging

Open pull requests against the linked repository to create a staging environment.

Title	Branch	Last Update Time	Status	Browse
<input type="checkbox"/> Enable fallback route	route	2020年5月25日 20:40:...	Ready	<a href="#">Browse</a>

ステージングされた変更から **Browse** をクリックして実際にページを確認してみると、about ページをリロードしても404ページが表示されないように改善されていることを確認できました。



あとは、プルリクエストをマージして production 環境にデプロイされるのを待つだけです。なお、プルリクエストがクローズされると、ステージング環境は自動的に削除されます。



以上、スケーラブルな構成の Vue.js アプリケーションがとても簡単に Azure でホスティングできることが確認できたのではないでしょうか？

ちなみに漫画の中にもあったように、Azure Static Web Apps は現在 Public Preview で無償で利用できます。

---

解説記事初出：

Vue.js で作るサイトを 無償の Azure Static Web Apps に CI/CD も含めて数分でホスティングする方法  
<https://qiita.com/miyake/items/de75b5034840e2687319>

## マンガでわかる Azure Static Web Apps

編集：日本マイクロソフト株式会社

漫画・ストーリー：湊川あい @luminatoll

15~21ページ解説：Kazuyuki Miyake @kazuyukimiyake



