

VSCode Conference Japan 2021

Bicep💪+VSCode 楽々 Azure Deploy

[VSCode Conference Japan 2021](#)

by Takekazu Omi(*@Baleen.Studio*)

2021/11/20 v1.0.0



Takekazu Omi @Baleen.Studio



自己紹介

近江 武一 [@takekazuomi](#)

- 所属 [JAZUG](#)、[baleen.studio](#) ([仲間を募集中](#))
- [GitHub](#)
 - Azure Container Apps の[サンプルプロジェクト](#)
 - [bicepのオレオレdevcontainer](#)
- Blog [kyrt.in](#) から [zenn.dev](#) へ移動 (したい)
 - [ARM template DSL、Bicep を使おう\(1\)](#)
 - [ARM template DSL、Bicep を使おう\(2\)](#)

今日の話

今日は、Bicep と VSCodeの話をしてします。

- そもそも、Bicep って何？
- VSCodeとBicepの関係
- Demo
- おまけ（時間があれば）

Bicepの誕生

- Azure では、VM、Webサーバー、データベースなど全てはリソース
- ARM Templateでは宣言的にリソース構成を定義できる

😓😓 ARM Template 😓😓

そこで 📌 Bicep 誕生 💪 @2020/9

Bicep の短い紹介



- 公式ドキュメント
 - <https://docs.microsoft.com/.../bicep/overview>
- MS Learn、Bicep 概要
 - <https://docs.microsoft.com/.../introduction-to-infrastructure-as-code-using-bicep/>
- ソースコード、GitHub
 - <https://github.com/Azure/bicep>

もう少し詳しく

- 宣言型の構文のドメイン固有言語 (DSL)
- 簡潔な構文
- 信頼性の高いタイプセーフ
- コード再利用のサポート
- ARM Templateに1:1で対応するトランスパイラ



Bicep の :-1

- 
 - ARM Template 直より 100 倍 楽 に書ける
 - 薄いラッパーなので、トラブルシューティングが楽
- 
 - ARM Template の根本的制約は解決できない
 - ↑ このあたりは別途

Bicepへの道

ARM Template の生産性が低い、どうすれば？ 🙄

- JSONの冗長性を避ける独自構文
 - DSLの導入
- 型を導入
 - IDE支援の強化、インテリセンス(=LanguageServerの実装)

※ ARM Template と Bicepの関係は、JS と TSに類似

型の導入

- 元々Azure Resource の型情報は、[azure-rest-api-specs](https://github.com/Azure/azure-rest-api-specs)で公開
 - swagger(Open API)の形式から、Bicep用の型情報を生成
 - <https://github.com/Azure/bicep-types-az>
 - 出来上がったものは、[nuget: Azure.Bicep.Types.Az](https://www.nuget.org/packages/Azure.Bicep.Types.Az) にある
 - Type情報の実体は、api spec から抜き出したJSONをリソースに埋込んだもの

Bicep LanguageServer

- <https://github.com/Azure/bicep/tree/main/src/Bicep.LanguageServer>
 - 型情報があるので、エディターでの自動補完の範囲が増えた
 - <https://microsoft.github.io/language-server-protocol/>

Bicep + VSCode

- Bicepの良さの半分はVCodeで出来ている
- Azure リソースの型情報を使って、VSCode上でLangage Serverを動かしてインテリセンスを実装
 - api specs が間違っていることがあるが、積極的に直す方針らしい 😊
 - <https://github.com/Azure/bicep/issues/4577>

Demo - IntelliSense

<https://github.com/Azure/bicep/tree/main/src/vscode-bicep>

1. Dot-property access
2. Resource property names & property values
3. List all available resource types
4. Snippets
5. Find all references, peek references
6. Outline view and breadcrumb view
7. Refactoring

Demo - devcontainerでの利用

- 現状 futures に対応していない
- ["Azure Bicep \(Community\)"](#) コンテナがあるが
 - azure cliか、azure poshがあれば足りる
 - 手でdevcontainer.jsonに、bicep extensions を入れる

```
"extensions": [  
  "ms-azuretools.vscode-bicep"  
],
```

新機能にも即時対応

Azure Native APIなので、新機能にも即時に対応

しかし、ドキュメントがなかったり、肝心の型情報が無かったりで結構辛い。

例: [Azure Container Apps](#)

最後に

⚡ [Tokyo Jazug Night](#)

- 11/25 20時、Bicep 0.5 の新機能

⚡ [ASK The Speaker](#)

- Twitter Spaces URL (TrackB)
- <https://twitter.com/i/spaces/1DXxyDBqqLVJM>



今回のコンテンツ

- GitHub [20211120-vscode-bicep](#)
- Slide share [20211120-vscode-bicep](#)
- Powered by [Marp](#)。ありがとうございました 👍

終

