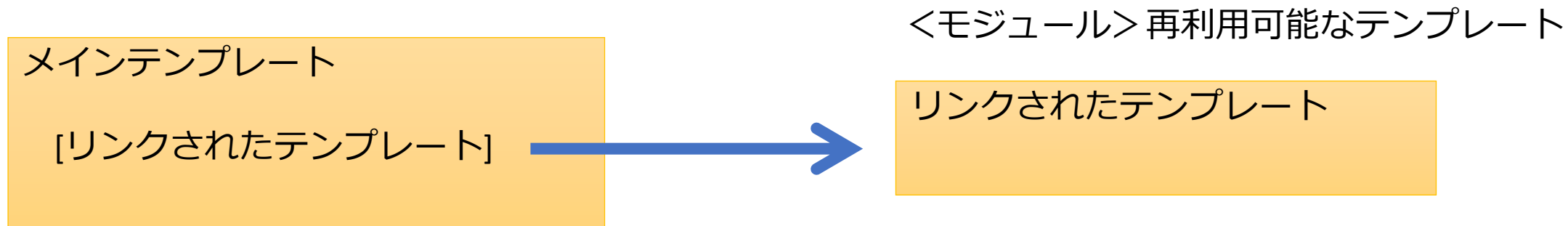


リンクされたテンプレート

大きなテンプレートを分割

リンクされたテンプレート



- テンプレートの分割、モジュール化のしくみ
- テンプレートを「メインテンプレート」と「リンクされたテンプレート」に分割
- メインテンプレート側で、リンクされたテンプレートの場所を指定



- 両方ともARMからアクセスできる場所（＝インターネット）に置くのが最もかんたん
- ローカルから、コマンドを使ってデプロイ

GitHubでホストする例

テンプレートを公開してもよい場合

メインテンプレートとリンクされたテンプレート両方をパブリックなGitHubリポジトリに格納

メインテンプレート

[リンクされたテンプレート]



template1.json

リンクされたテンプレート

[リンクされたテンプレート] の実際の記述

```
{
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2020-10-01",
  "name": "linkedTemplate",
  "properties": {
    "mode": "Incremental",
    "templateLink": {
      "uri": "[url(deployment().properties.templateLink.uri, 'template1.json')]",
      "contentVersion": "1.0.0.0"
    }
  }
}
```

親テンプレートの URI に基づいて、
別のテンプレートにリンク

問題点

- ポータルを使用して再デプロイ できない



メインテンプレート

[リンクされたテンプレート]



template1.json

リンクされたテンプレート

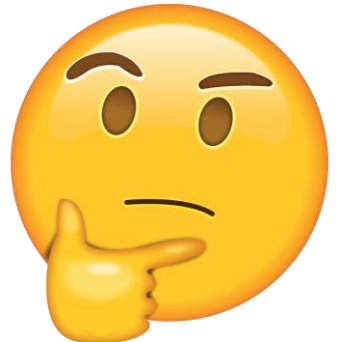
[リンクされたテンプレート] の実際の記述

```
{
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2020-10-01",
  "name": "linkedTemplate",
  "properties": {
    "mode": "Incremental",
    "templateLink": {
      "uri": "https://raw.githubusercontent.com/user1/repo1/main/template1.json",
      "contentVersion": "1.0.0.0"
    }
  }
}
```

リンクされたテンプレートのURLを記入

問題点

- リンクされたテンプレートのURLは、変わる可能性がある
 - テンプレートを別の場所に移動した場合
 - フォークした場合
- リンクされたテンプレートのURLが変わったら、メインテンプレート内のURLも更新しなければならない



Azure Blob Storageで ホストする例(1)

Blobのコンテナを公開してもよい場合

ストレージアカウント storageaccount1

Blobコンテナ container1 <匿名パブリック読み取りアクセス：BLOB>

公開

メインテンプレート

[リンクされたテンプレート]

template1.json

リンクされたテンプレート



[リンクされたテンプレート] の実際の記述

```
{
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2020-10-01",
  "name": "linkedTemplate",
  "properties": {
    "mode": "Incremental",
    "templateLink": {
      "uri": " [url(deployment().properties.templateLink.uri, 'template1.json')] ",
      "contentVersion": "1.0.0.0"
    }
  }
}
```

リンクされたテンプレートのURLを指定

問題点

- ポータルを使用して再デプロイ できない



ストレージアカウント storageaccount1

Blobコンテナ container1 <匿名パブリック読み取りアクセス：BLOB>

公開

メインテンプレート

[リンクされたテンプレート]

template1.json

リンクされたテンプレート



[リンクされたテンプレート] の実際の記述

```
{
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2020-10-01",
  "name": "linkedTemplate",
  "properties": {
    "mode": "Incremental",
    "templateLink": {
      "uri": "https://storageaccount1.blob.core.windows.net/container1/template1.json",
      "contentVersion": "1.0.0.0"
    }
  }
}
```

リンクされたテンプレートのURLを指定

相対パスで書きたいが・・・だめ

問題点

- リンクされたテンプレートのURLは、変わる可能性がある
 - テンプレートを別の場所に移動した場合
- リンクされたテンプレートのURLが変わったら、メインテンプレート内のURLも更新しなければならない



Azure Blob Storageで ホストする例(2)

テンプレートを公開したくない場合

AZ-400ラボ13で紹介されている方式

ストレージアカウント storageaccount1

Blobコンテナ container1 <プライベート>

非公開

メインテンプレート

[リンクされたテンプレート]

template1.json

リンクされたテンプレート

[リンクされたテンプレート] の実際の記述

```
{
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2020-10-01",
  "name": "linkedTemplate",
  "properties": {
    "mode": "Incremental",
    "templateLink": {
      "uri": "https://storageaccount1.blob.core.windows.net/container1/template1.json?... ",
      "contentVersion": "1.0.0.0"
    }
  }
}
```

リンクされたテンプレートの
SAS URLを生成し、
メインテンプレートに書き込む

?以降はSASトークン

問題点

- **SAS URL**を生成しなければならない
- メインテンプレートに、生成した**SAS URL**を書き込まなければならない
- SAS URLの有効期限が過ぎてからデプロイする場合、再び上記2点を実行しなければならない



Azure Blob Storageで ホストする例(3)

テンプレートを公開したくない場合

Azure ドキュメントで紹介されている方式

ストレージアカウント storageaccount1

Blobコンテナ container1 <プライベート>

非公開

メインテンプレート

[リンクされたテンプレート]

template1.json

リンクされたテンプレート

[リンクされたテンプレート] の実際の記述

```
{
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2020-10-01",
  "name": "linkedTemplate",
  "properties": {
    "mode": "Incremental",
    "templateLink": {
      "uri": "concat(url(deployment().properties.templateLink.uri, 'template1.json'),
        parameters('containerSasToken'))",
      "contentVersion": "1.0.0.0"
    }
  }
}
```

メインテンプレートのベースURL
+ リンクされたテンプレートの相対パス
+ SASトークン (パラメータで指定)

token=<リンクされたURLのSASトークンを生成>

parameters='{ "containerSasToken": { "value": "?"\$token" } }'

az deployment group create ¥

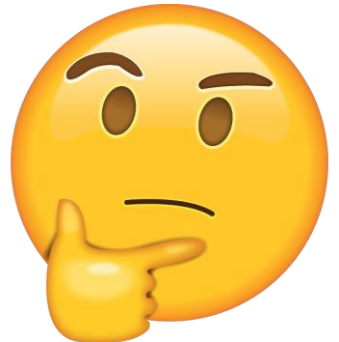
--resource-group ExampleGroup ¥

--template-uri \$url?\$token ¥

--parameters \$parameters

問題点

- **SAS トークン**を生成しなければならない
- **SAS トークン**をデプロイのコマンドにパラメータとして指定しなければならない
- **SAS トークン**の有効期限が過ぎたてからデプロイする場合、再びSASトークンを生成しなければならない



ローカルからのデプロイ

テンプレートをローカルで開発し、インターネットに置かずにデプロイしたい

ローカルPC

メインテンプレート

[リンクされたテンプレート]

インターネット

template1.json

リンクされたテンプレート



- メインテンプレートは、ローカルに置くこともできる
 - コマンドによって読み取られ、ARMに送信される。
- リンクテンプレートは、ARMからアクセスできる場所（=インターネット）に配置しなければならない



テンプレートスペック

2021/5/7 一般提供開始

ここまで説明した問題点を解消できる

- 現在、GitHub リポジトリまたはストレージ アカウントにテンプレートがある場合に、テンプレートを共有して使用しようとすると、**いくつかの問題**が発生します。
- テンプレートをデプロイするには、**テンプレートへのパブリック アクセスを許可するか、SAS トークンを使用してアクセスを管理する**必要があります。
- **テンプレート スペック**により、テンプレートの共有が簡単になります。

ローカルPCなどの任意の環境

GitHubやBlobコンテナが不要！

メインテンプレート

[リンクされたテンプレート]



template1.json

リンクされたテンプレート

[リンクされたテンプレート] の実際の記述

```
{
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2020-06-01",
  "name": "linkedTemplate",
  "properties": {
    "mode": "Incremental",
    "templateLink": {
      "relativePath": "template1.json",
    }
  }
}
```

相対パスで指定可能！

テンプレートスペック作成

az **ts** create ¥

--name tspec1 ¥

--version "1.0" ¥

--resource-group rg1 ¥

--location eastus ¥

--template-file helloworldparent.json

リンクされたテンプレートは指定不要。
メインテンプレートから自動的にた
どってくれる

テンプレートスペックのIDを確認

az **ts** show -g rg1 --name tspec1

IDの指定が必要なのがちょっ
とイケてないが・・・今後
改善されるかも？

テンプレートスペックのデプロイ

az deployment group create

--name tspec1 ¥

--resource-group rg1 ¥

--template-spec **テンプレートスペックのID**