

Arrancando curso 2021-2022 con fuerza: Bicep! 💪

¿Qué es Azure Bicep?

Es un proyecto de Microsoft

Se trata de un DSL para desplegar recursos de Azure de forma declarativa

Su finalidad: simplificar drásticamente la creación de soluciones de IaC en Azure y proveer una abstracción transparente sobre ARM y plantillas ARM.

Principales ventajas de Bicep

- Soporte para todos los tipos de recursos y versiones de API **de manera inmediata**
- **Sintaxis mucho más simple...** sobre todo comparando con plantillas ARM
- Tipado, IntelliSense y validación de sintaxis (con VS Code)
- Modularidad
- Integración con servicios de Azure (policies, template specs, blueprints)
- **No tenemos que manejar el estado**, Azure lo hará por nosotros

Comparando Bicep y plantillas ARM

```
1 param serverName string = uniqueString('sql', resourceGroup().id)
2 param sqlDBName string = 'SampleDB'
3 param location string = resourceGroup().location
4 param administratorLogin string
5
6 @secure()
7 param administratorLoginPassword string
8
9 resource server 'Microsoft.Sql/servers@2019-06-01-preview' = {
10   name: serverName
11   location: location
12   properties: {
13     administratorLogin: administratorLogin
14     administratorLoginPassword: administratorLoginPassword
15   }
16 }
17
18 resource sqlDB 'Microsoft.Sql/servers/databases@2020-08-01-preview' = {
19   name: '${server.name}/${sqlDBName}'
20   location: location
21   sku: {
22     name: 'Standard'
23     tier: 'Standard'
24   }
25 }
26
```

```
1 {
2   "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTem
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "serverName": {
6       "type": "string",
7       "defaultValue": "[uniqueString('sql', resourceGroup().id)]"
8     },
9     "sqlDBName": {
10      "type": "string",
11      "defaultValue": "SampleDB"
12    },
13    "location": {
14      "type": "string",
15      "defaultValue": "[resourceGroup().location]"
16    },
17    "administratorLogin": {
18      "type": "string"
19    },
20    "administratorLoginPassword": {
21      "type": "secureString"
22    }
23  },
24  "functions": [],
25  "resources": [
26    {
27      "type": "Microsoft.Sql/servers",
28      "apiVersion": "2019-06-01-preview",
29      "name": "[parameters('serverName')]",
30      "location": "[parameters('location')]",
31      "properties": {
32        "administratorLogin": "[parameters('administratorLogin')]",
33        "administratorLoginPassword": "[parameters('administratorLoginPassword')]"
34      }
35    },
36  ]
37 }
```

Visión general

Project Bicep

Simple declarative language to provision infrastructure on Azure

INTUITIVE

Easy to read as it is to author

TRANSPILES TO ARM TEMPLATES

Leverage ARM template knowledge and investments

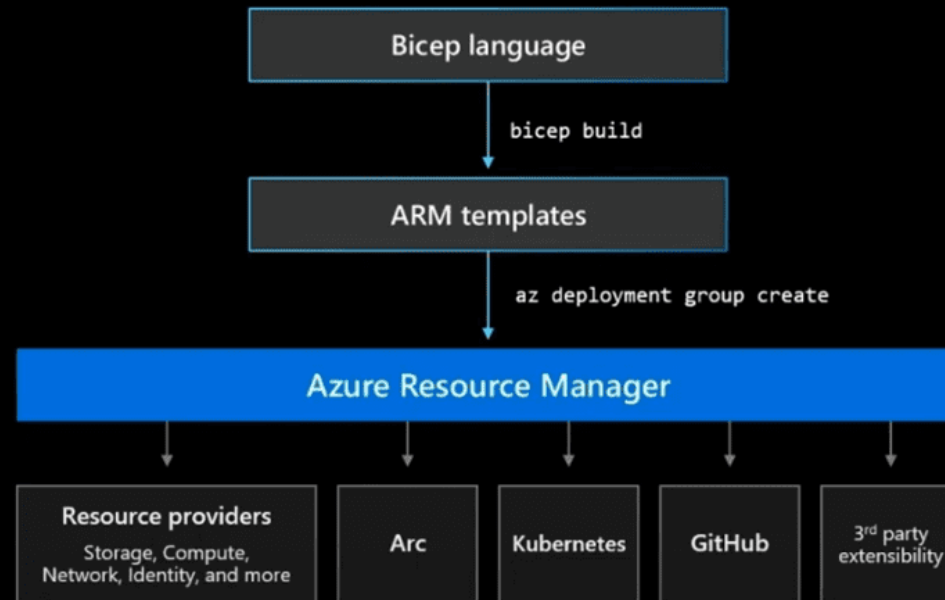
MODULAR

Abstract common blocks of config into reusable elements

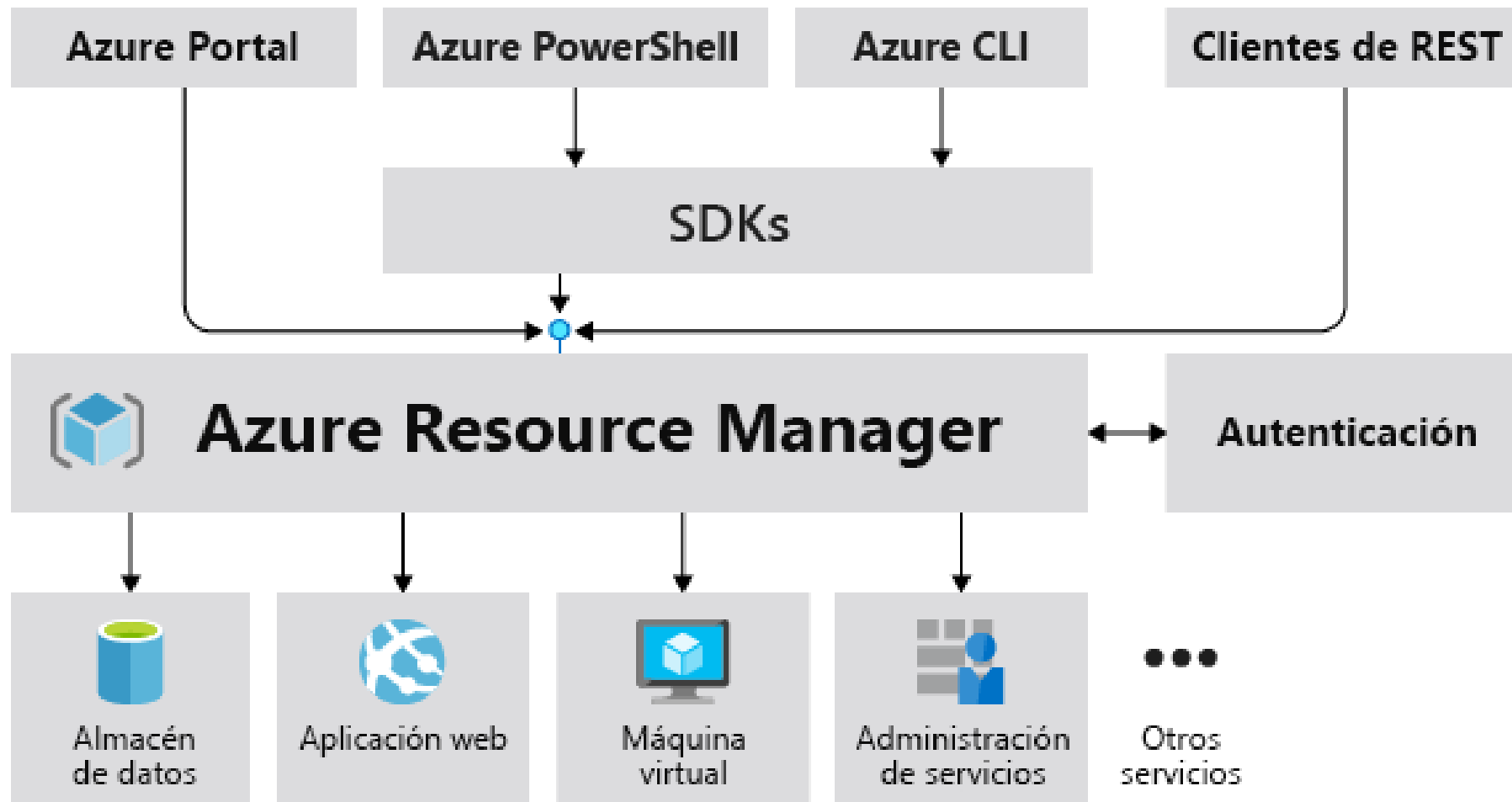
OPEN SOURCE

Transparency and community

aka.ms/Bicep



Azure Resource Manager (ARM)



¿Por qué ahora?

- Se consideró crear Bicep como **una "revisión" a el lenguaje de las plantillas ARM existentes** (por su audiencia objetivo)
- Bicep tiene como audiencia objetivo aquellos **que aún no han adoptado soluciones de IaC o tienen plantillas ARM en uso**
- Cada uno que use las herramientas y lenguajes que más le gusten

¿Que pasa con otras ofertas de IaC?

Terraform, Pulumi y otras soluciones de IaC **seguirán estando ahí y no hay necesidad de cambiar a Bicep.**

Microsoft y la comunidad se encargarán de que funcionen sobre Azure lo mejor que puedan.

¿Que pasa ahora con las plantillas ARM?

Siguen y seguirán funcionando como siempre. **No hay necesidad de hacer cambios**

Se continua el soporte del lenguaje de las plantillas ARM

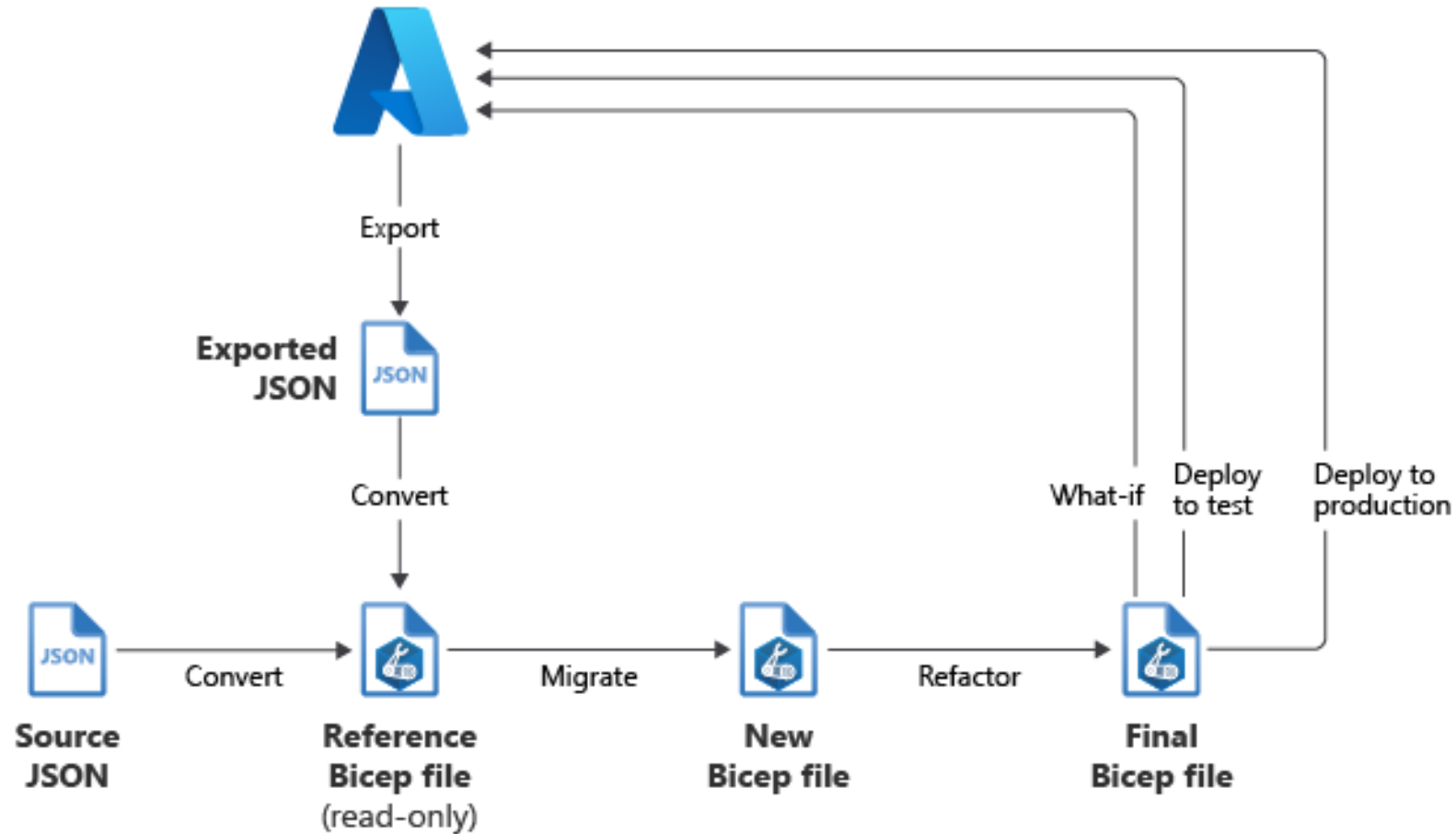
Al fin y al cabo, **Bicep compila a estas plantillas ARM en JSON** y este es enviado a Azure para su despliegue

Empezando con Bicep...

Dos escenarios:

- Partimos desde plantillas ARM que estamos usando
- Partimos de cero

Estoy usando plantillas ARM... ¿como paso a Bicep?



Estoy usando plantillas ARM... ¿como paso a Bicep?

Con AZ CLI:

```
az bicep decompile --file main.json
```

Si tenemos la CLI instalada:

```
bicep decompile --file main.json
```

Desde cero...

Bastante simple empezar con Bicep, solo necesitaremos [Azure CLI](#) y, opcionalmente, [CLI de Bicep](#) y la [extensión de VS Code](#)

Este tooling nos ayudará bastante con la creación/modificación de ficheros Bicep

Tutoriales en [Microsoft Learn](#) y "[Bicep Playground](#)" para empezar a probar sin necesidad de instalar nada

Más recursos

Para CI/CD

- [Bicep GitHub Action](#)
- [Bicep Tasks extension for Azure Pipelines](#)

Otros:

- [Bicep Language Service support in Neovim](#)
- [Bicep PowerShell Module](#)

Veamos un poco como funciona

¡Gracias por asistir! 😊

Recursos: bit.ly/bicep_demo

Presentación: bit.ly/ppt-bicep