Bicep – Infrastructure as Code Masterclass – part II

Esther Barthel

Freek Berson

10TH ANNIVERSARY EDITION

Experts Live Netherlands

Esther Barthel

Solutions Architect and owner @ cognition IT
@virtuEs_IT

- Loves workflows, DevOps, Bicep & scripting
- Women in Tech Mentorship advocate
- Microsoft MVP RDS/CDM since 2017

Freek Berson

Principal Outbound Product Manager
Parallels (Alludo)
@fberson

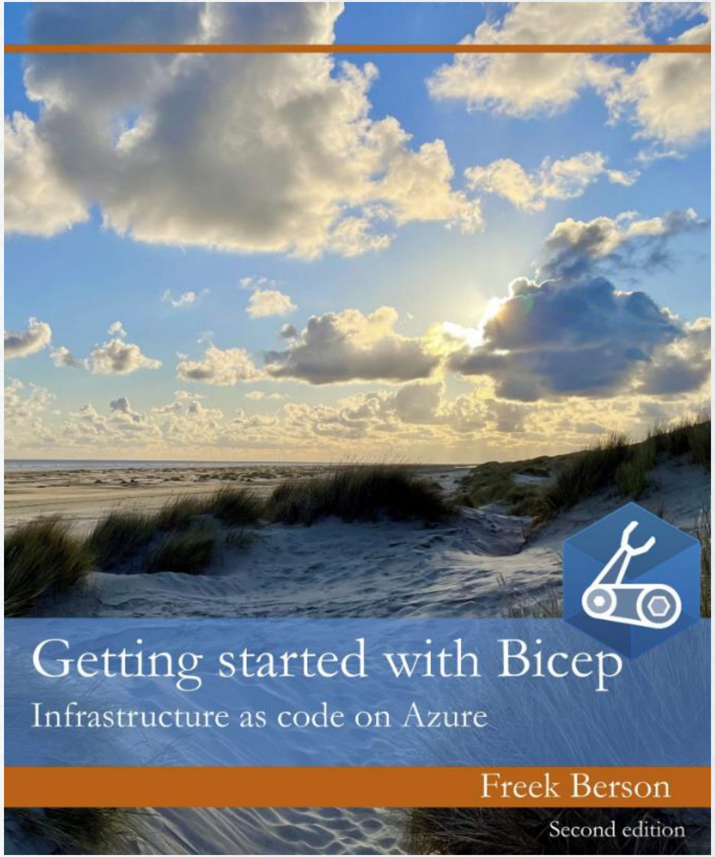
- Fell in love with Project Bicep since its inception.
- Book author: ‘Getting started with Bicep’
- Microsoft MVP on RDS since 2011

Delta-N Connecting the Cloud cegeka ARROW LIQUIT INSPARK Microsoft

DEVOPS

Tweet to win!

@fberson
@virtuEs_IT
#ExpertsLiveNL



Getting started with Bicep

Infrastructure as code on Azure

Freek Berson

Second edition

Agenda

~~Part I: Focus on how to get started (13.00u)~~

- ~~Introducing ARM, IaC, Templates~~
- ~~Demo: Bicep & VSCode~~

Part II: Advanced templates & deployment (14.00)

- Demo: Advanced Bicep Language capabilities
- Deployment methods

DEVOPS

Bicep to empower your ARM!

Mark Russinovich
@markrussinovich

We're working on an open source domain specific language for ARM codenamed Bicep that will greatly simplify Azure declarative modelling: "Microsoft flexes Bicep to strengthen ARM"

What is Bicep?

".. Bicep is a **Domain Specific Language** (DSL) for deploying Azure resources declaratively. It aims to **drastically simplify the authoring experience** with a cleaner syntax and better support for modularity and code re-use.

Bicep is a transparent abstraction over ARM and ARM templates."

What is Bicep?

Intuitive

Easy to read and to author

Transpiles to ARM Templates

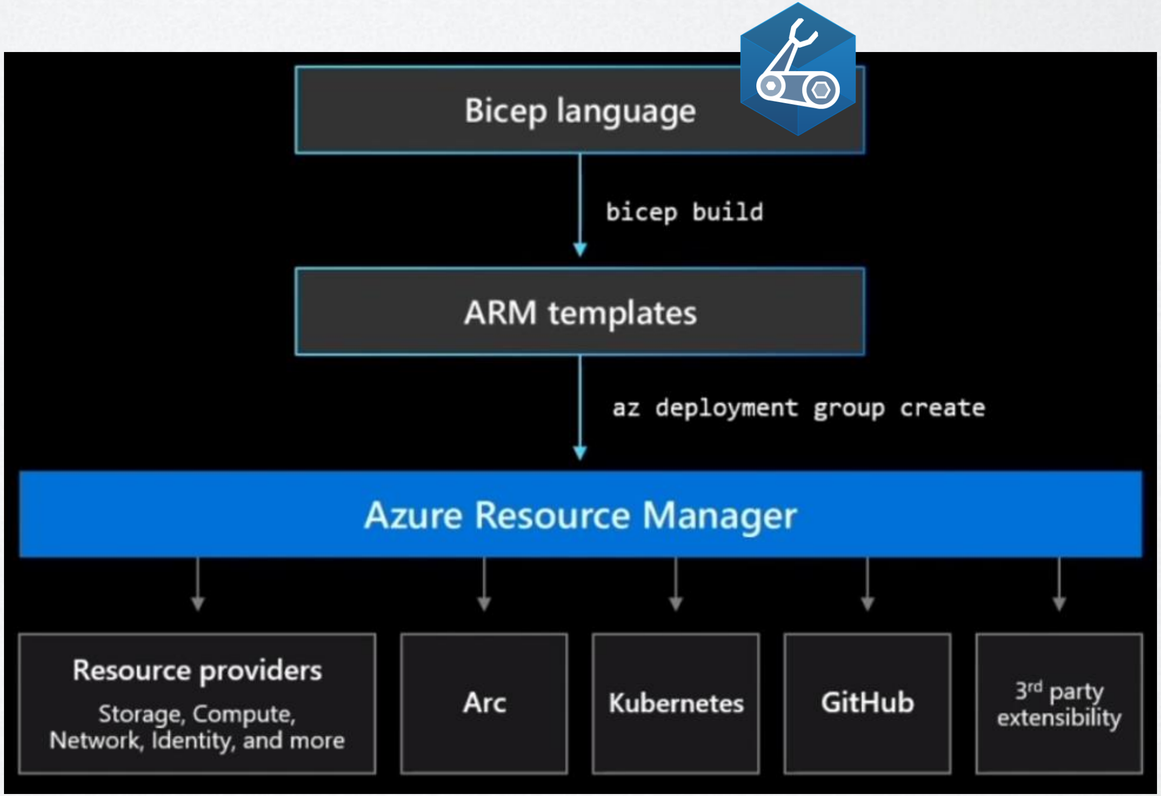
Leverage ARM template knowledge and investments

Modular

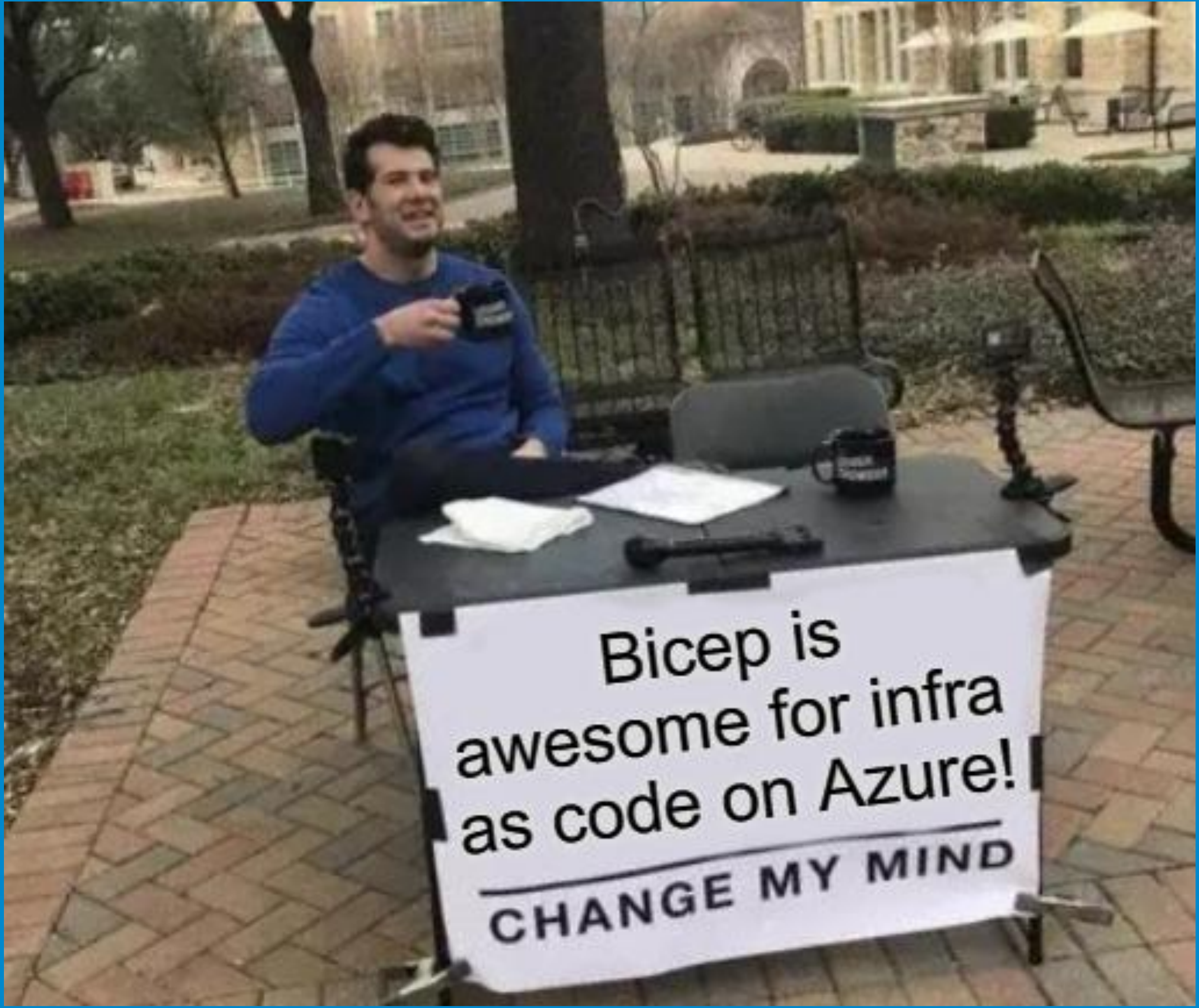
Abstract common blocks of code into reusable parts

Open Source

Transparency and community



DEMO!
Advanced Bicep code

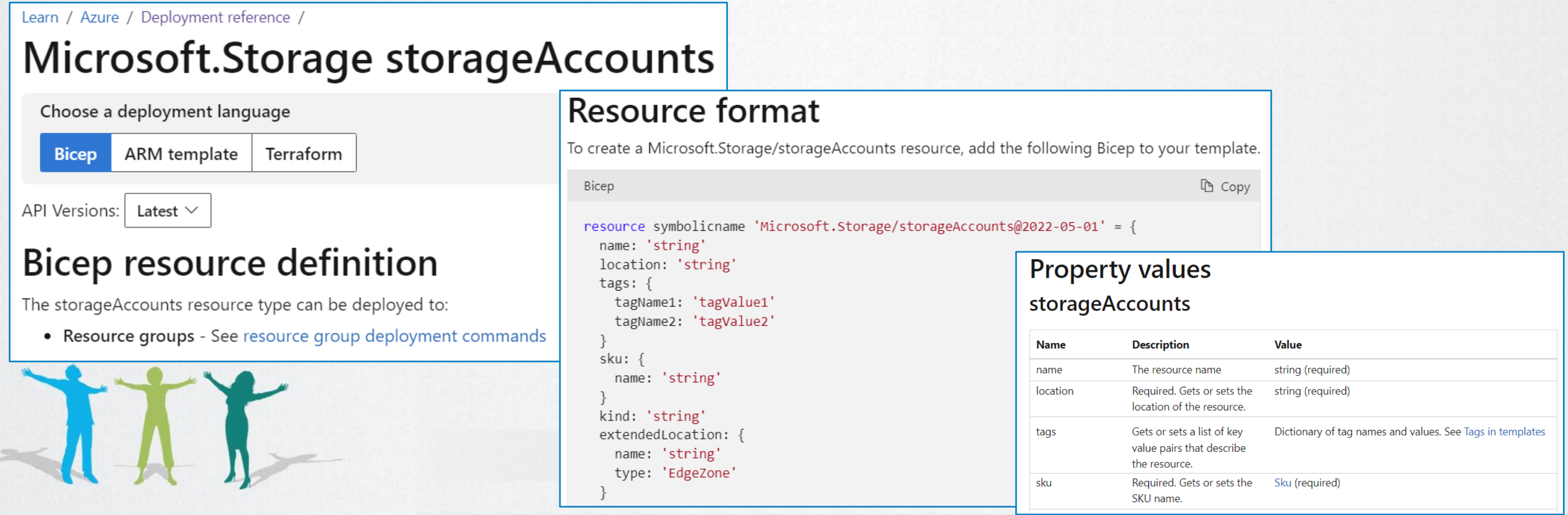


DEVOPS

Deployment reference

If you know the resource type, you can go directly to it with the following URL format:
https://docs.microsoft.com/azure/templates/{provider-namespace}/{resource-type}

for storage accounts, go to:
https://docs.microsoft.com/azure/templates/microsoft.storage/storageaccounts



Learn / Azure / Deployment reference /

Microsoft.Storage storageAccounts

Choose a deployment language

API Versions: Latest

Bicep resource definition

The storageAccounts resource type can be deployed to:

- Resource groups - See resource group deployment commands

Resource format

To create a Microsoft.Storage/storageAccounts resource, add the following Bicep to your template.

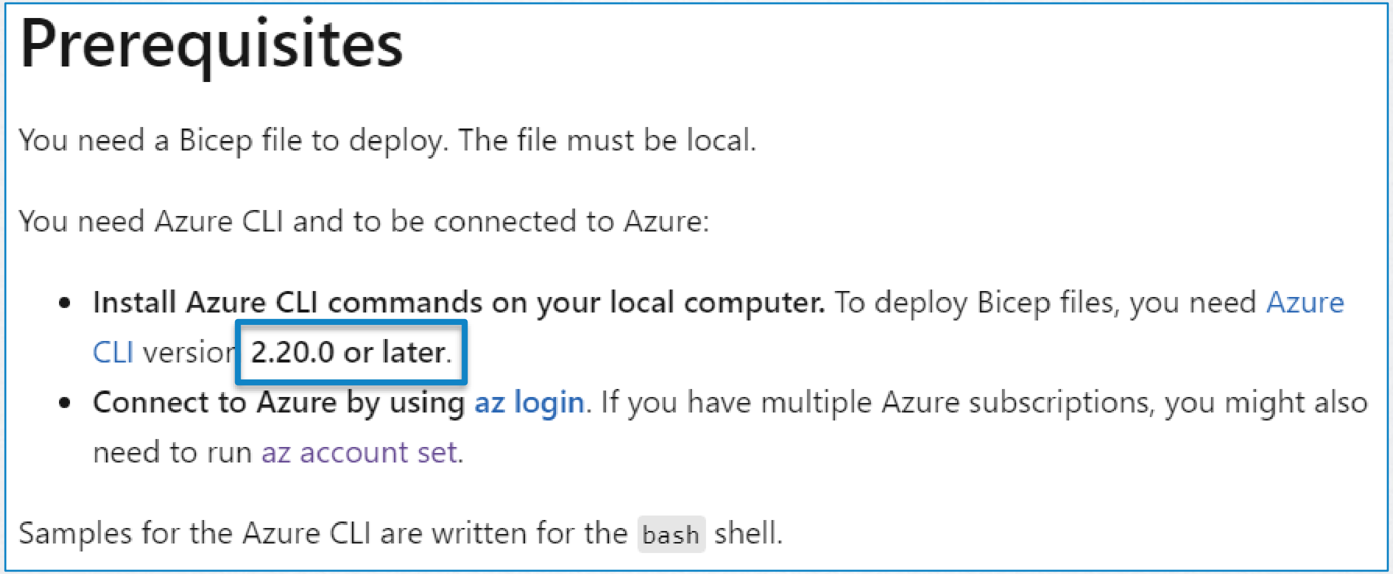
```
resource symbolicname 'Microsoft.Storage/storageAccounts@2022-05-01' = {
  name: 'string'
  location: 'string'
  tags: {
    tagName1: 'tagValue1'
    tagName2: 'tagValue2'
  }
  sku: {
    name: 'string'
  }
  kind: 'string'
  extendedLocation: {
    name: 'string'
    type: 'EdgeZone'
  }
```

Property values

storageAccounts

Name	Description	Value
name	The resource name	string (required)
location	Required. Gets or sets the location of the resource.	string (required)
tags	Gets or sets a list of key value pairs that describe the resource.	Dictionary of tag names and values. See Tags in templates
sku	Required. Gets or sets the SKU name.	Sku (required)

Deploy resources w/ Bicep & Azure CLI



Prerequisites

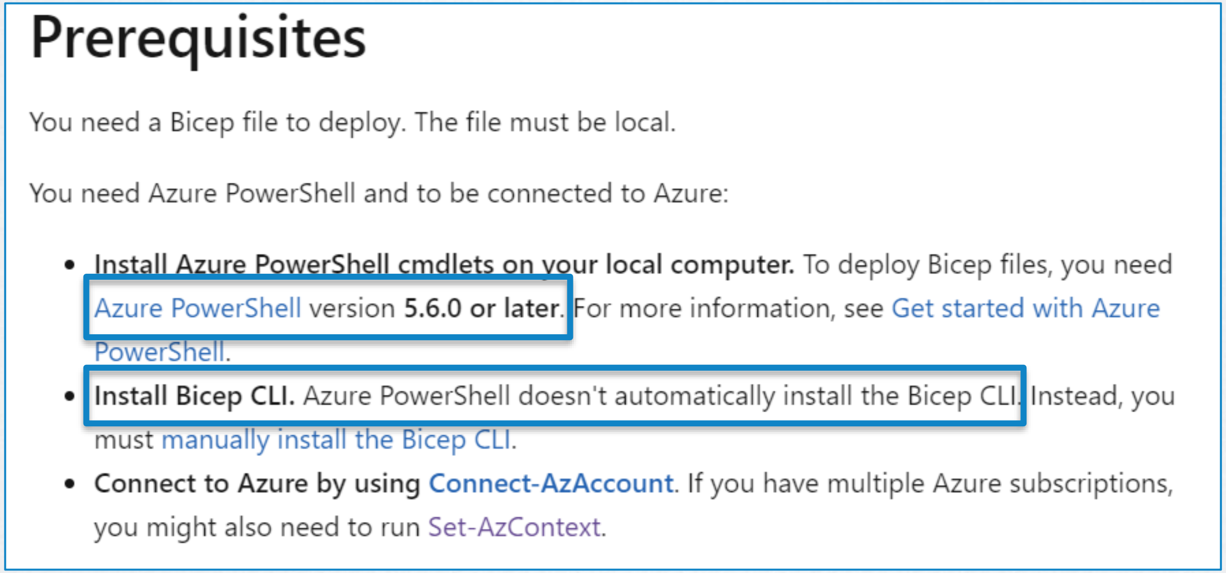
You need a Bicep file to deploy. The file must be local.

You need Azure CLI and to be connected to Azure:

- **Install Azure CLI commands on your local computer.** To deploy Bicep files, you need Azure CLI version 2.20.0 or later.
- **Connect to Azure by using az login**. If you have multiple Azure subscriptions, you might also need to run az account set.

Samples for the Azure CLI are written for the `bash` shell.

Deploy resources w/ Bicep & PowerShell

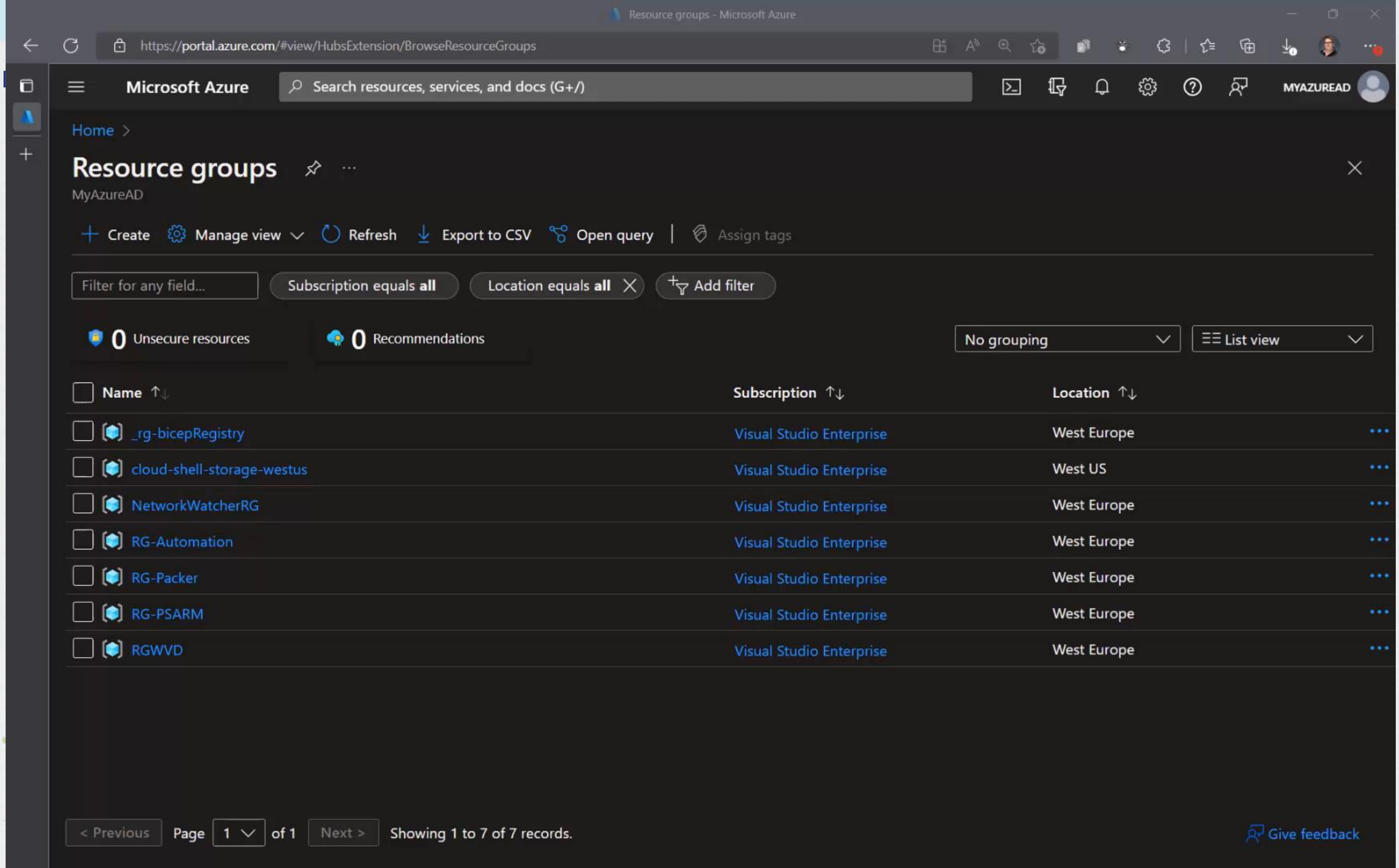


Prerequisites

You need a Bicep file to deploy. The file must be local.

You need Azure PowerShell and to be connected to Azure:

- **Install Azure PowerShell cmdlets on your local computer.** To deploy Bicep files, you need Azure PowerShell version **5.6.0 or later**. For more information, see Get started with Azure PowerShell.
- **Install Bicep CLI.** Azure PowerShell doesn't automatically install the Bicep CLI. Instead, you must manually install the Bicep CLI.
- **Connect to Azure by using Connect-AzAccount.** If you have multiple Azure subscriptions, you might also need to run Set-AzContext.


Resource groups - Microsoft Azure
https://portal.azure.com/#view/HubsExtension/BrowseResourceGroups
Microsoft Azure
Search resources, services, and docs (G+/)
MYAZUREAD
Home >
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Filter for any field...
Add filter

Resource groups

MyAzureAD

Subscription equals **all**

Location equals **all**

0 Unsecure resources

0 Recommendations

No grouping

List view

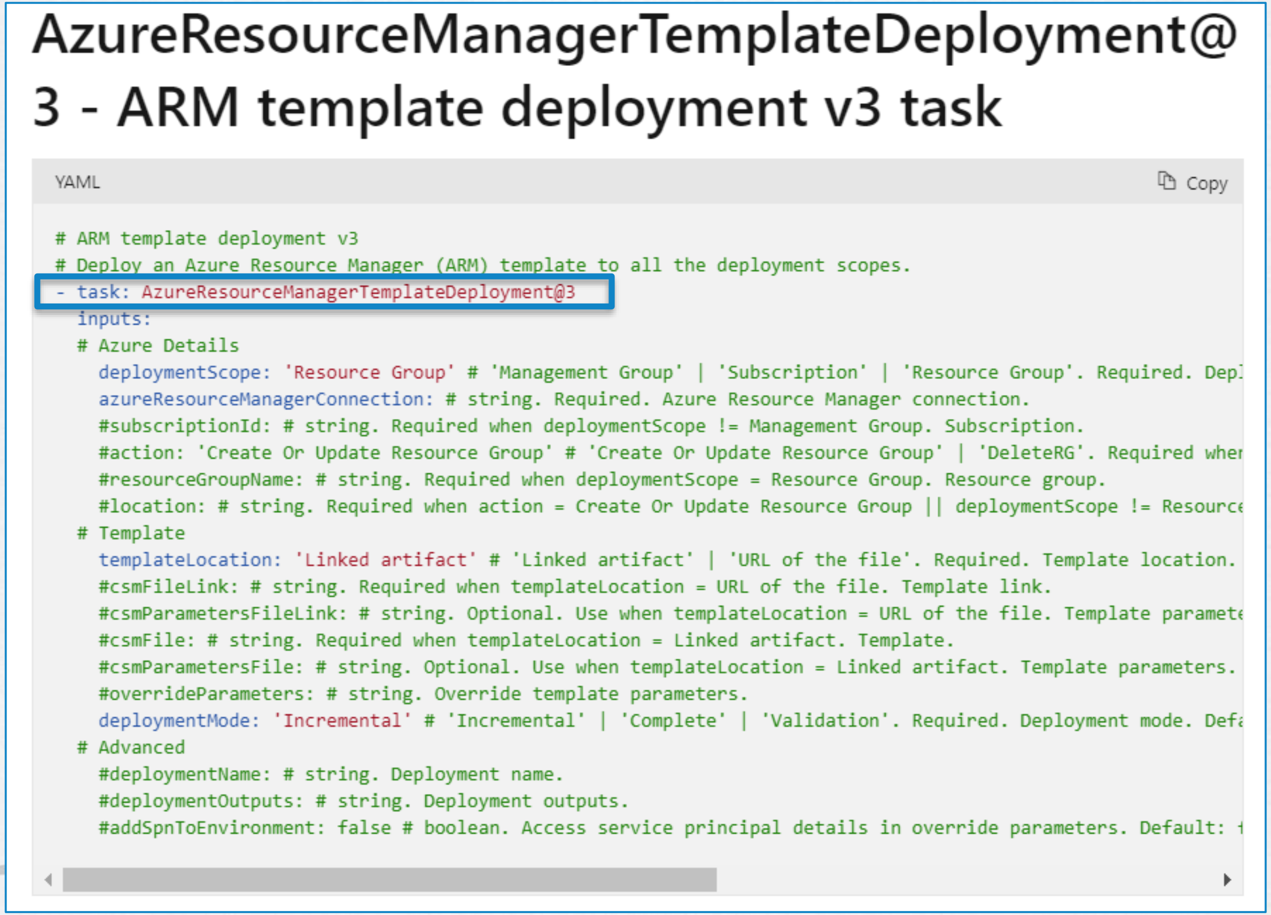
Name	Subscription	Location
_rg-bicepRegistry	Visual Studio Enterprise	West Europe
cloud-shell-storage-westus	Visual Studio Enterprise	West US
NetworkWatcherRG	Visual Studio Enterprise	West Europe
RG-Automation	Visual Studio Enterprise	West Europe
RG-Packer	Visual Studio Enterprise	West Europe
RG-PSARM	Visual Studio Enterprise	West Europe
RGWVD	Visual Studio Enterprise	West Europe

< Previous Page 1 of 1 Next >

Showing 1 to 7 of 7 records.

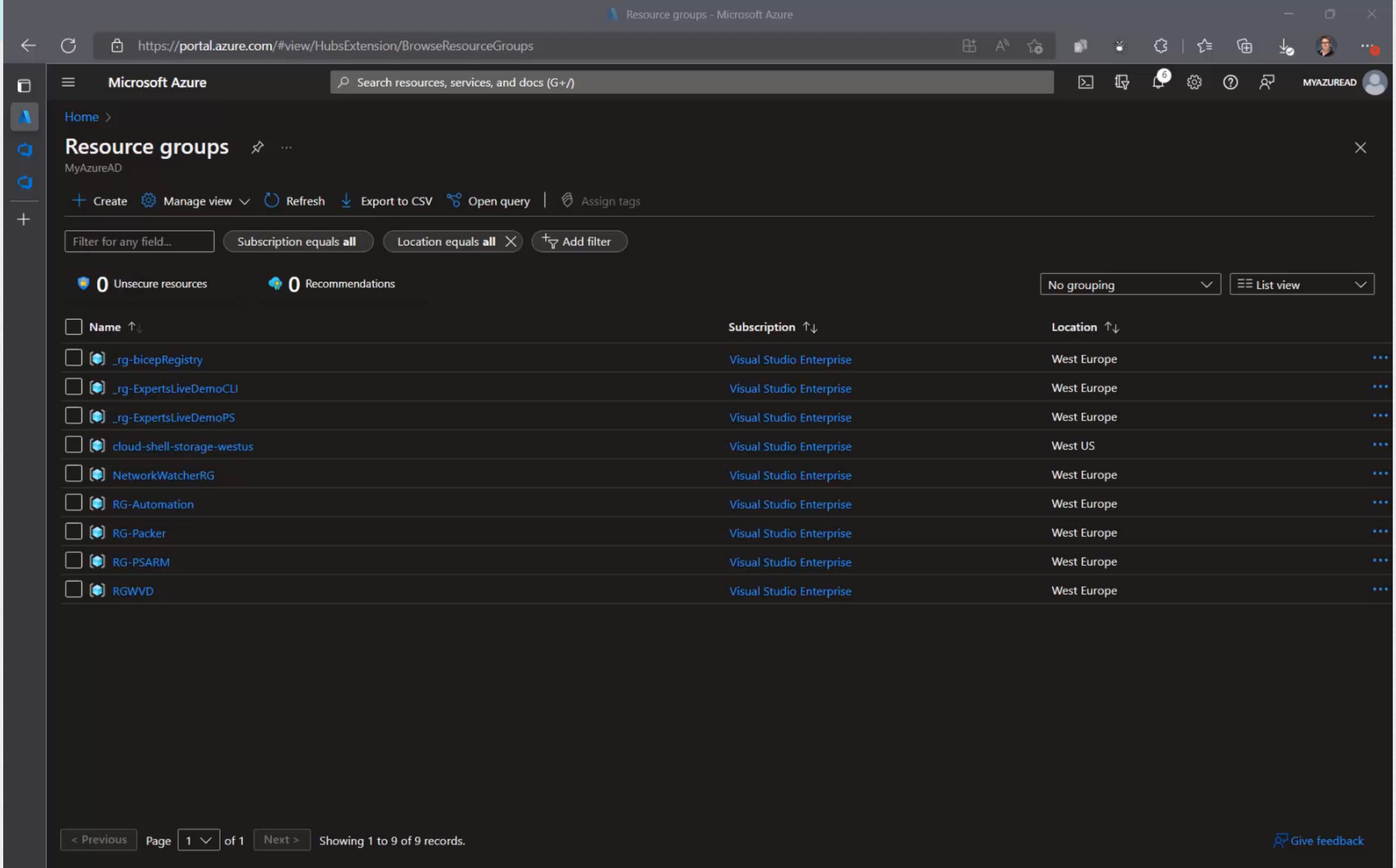
DEVOPS

Deploy resources w/ Bicep & Azure DevOps



AzureResourceManagerTemplateDeployment@3 - ARM template deployment v3 task

```yaml
# ARM template deployment v3
# Deploy an Azure Resource Manager (ARM) template to all the deployment scopes.
- task: AzureResourceManagerTemplateDeployment@3
  inputs:
  # Azure Details
    deploymentScope: 'Resource Group' # 'Management Group' | 'Subscription' | 'Resource Group'. Required. Dep
    azureResourceManagerConnection: # string. Required. Azure Resource Manager connection.
    #subscriptionId: # string. Required when deploymentScope != Management Group. Subscription.
    #action: 'Create Or Update Resource Group' # 'Create Or Update Resource Group' | 'DeleteRG'. Required whe
    #resourceGroupName: # string. Required when deploymentScope = Resource Group. Resource group.
    #location: # string. Required when action = Create Or Update Resource Group || deploymentScope != Resource
  # Template
    templateLocation: 'Linked artifact' # 'Linked artifact' | 'URL of the file'. Required. Template location.
    #csmFileLink: # string. Required when templateLocation = URL of the file. Template link.
    #csmParametersFileLink: # string. Optional. Use when templateLocation = URL of the file. Template paramete
    #csmFile: # string. Required when templateLocation = Linked artifact. Template.
    #csmParametersFile: # string. Optional. Use when templateLocation = Linked artifact. Template parameters.
    #overrideParameters: # string. Override template parameters.
    deploymentMode: 'Incremental' # 'Incremental' | 'Complete' | 'Validation'. Required. Deployment mode. Defa
  # Advanced
    #deploymentName: # string. Deployment name.
    #deploymentOutputs: # string. Deployment outputs.
    #addSpnToEnvironment: false # boolean. Access service principal details in override parameters. Default: 
```



Resource groups

MyAzureAD

0 Unsecure resources

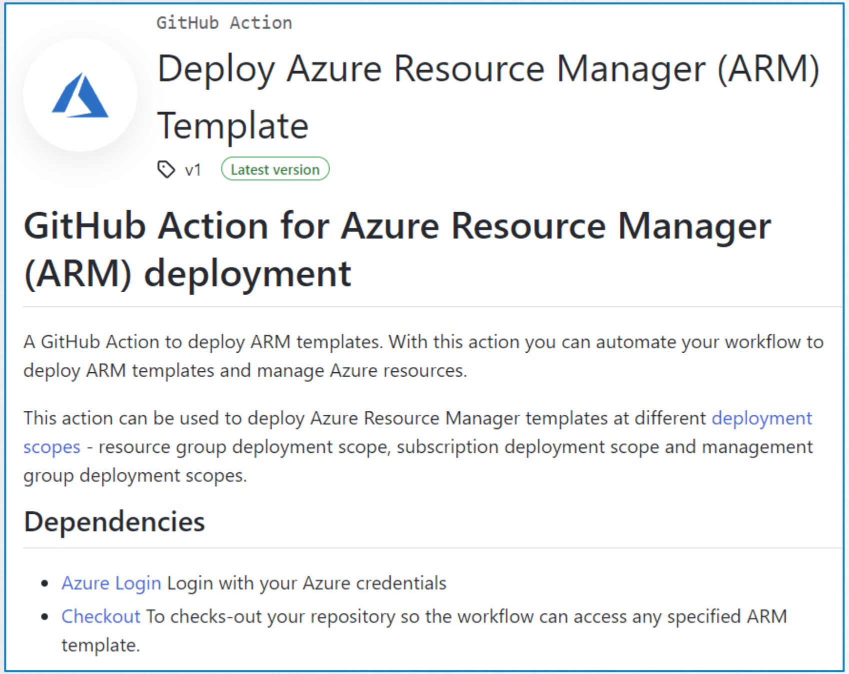
0 Recommendations

Name	Subscription	Location
_rg-bicepRegistry	Visual Studio Enterprise	West Europe
_rg-ExpertsLiveDemoCLI	Visual Studio Enterprise	West Europe
_rg-ExpertsLiveDemoPS	Visual Studio Enterprise	West Europe
cloud-shell-storage-westus	Visual Studio Enterprise	West US
NetworkWatcherRG	Visual Studio Enterprise	West Europe
RG-Automation	Visual Studio Enterprise	West Europe
RG-Packer	Visual Studio Enterprise	West Europe
RG-PSARM	Visual Studio Enterprise	West Europe
RGWVD	Visual Studio Enterprise	West Europe

Page 1 of 1 Showing 1 to 9 of 9 records.

DEVOPS

Deploy resources w/ Bicep & GitHub Actions



GitHub Action

Deploy Azure Resource Manager (ARM) Template

v1 Latest version

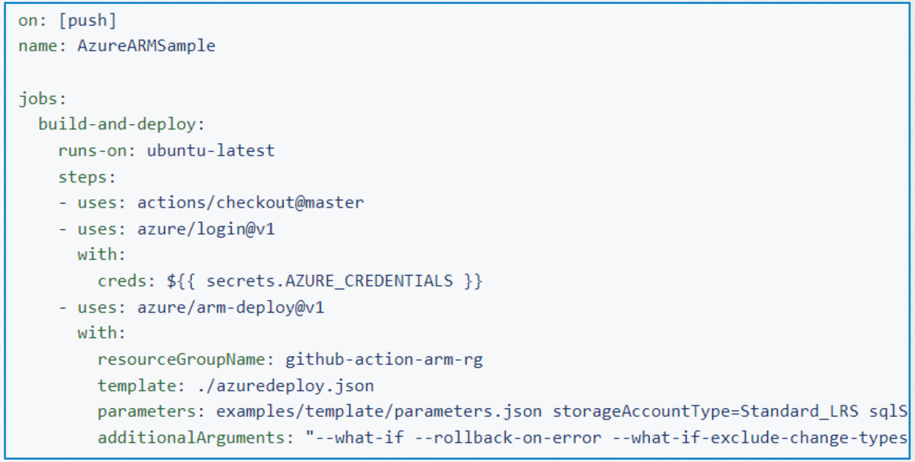
GitHub Action for Azure Resource Manager (ARM) deployment

A GitHub Action to deploy ARM templates. With this action you can automate your workflow to deploy ARM templates and manage Azure resources.

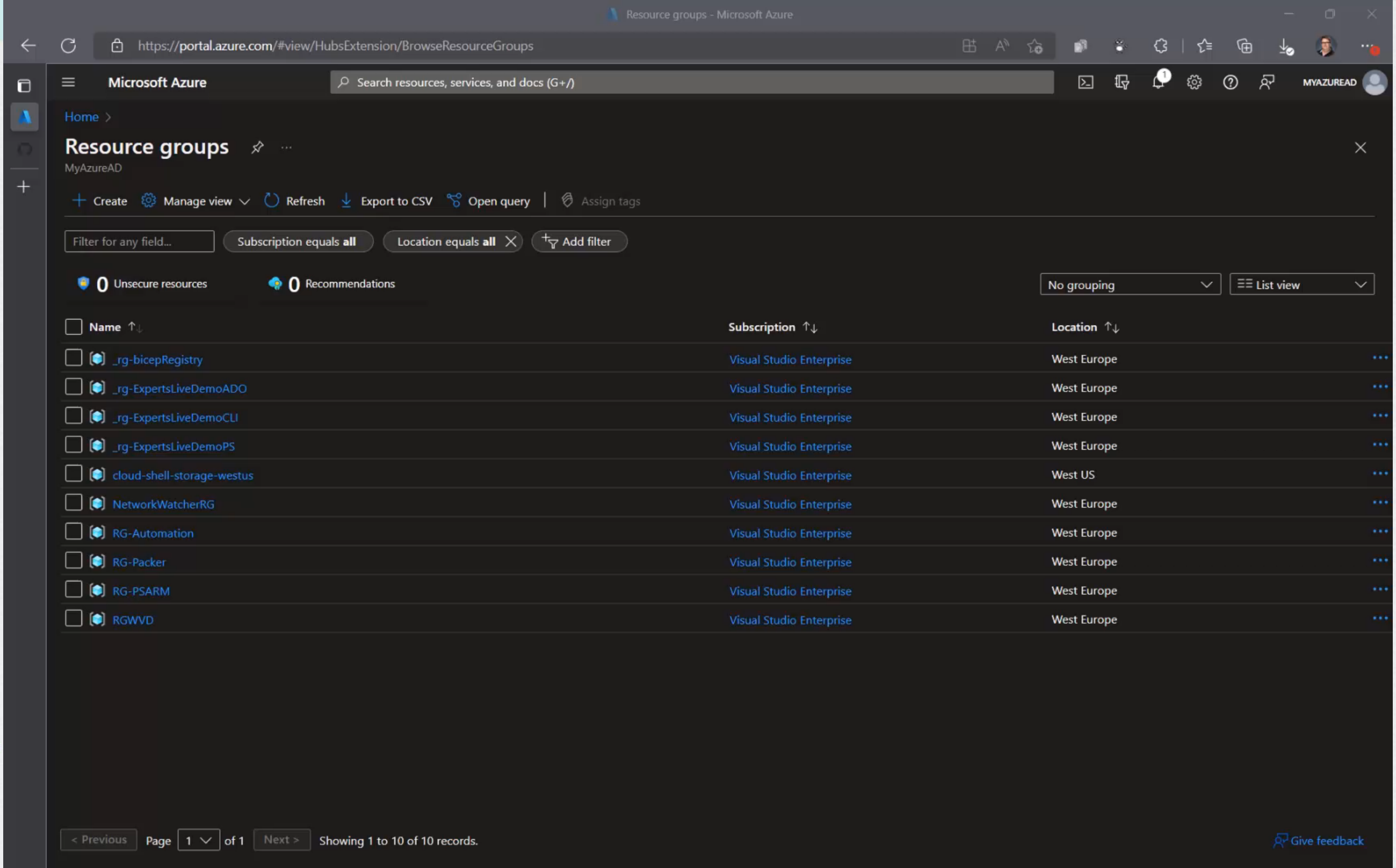
This action can be used to deploy Azure Resource Manager templates at different deployment scopes - resource group deployment scope, subscription deployment scope and management group deployment scopes.

Dependencies

- Azure Login Login with your Azure credentials
- Checkout To checks-out your repository so the workflow can access any specified ARM template.



```
on: [push]
name: AzureARMSample

jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@master
    - uses: azure/login@v1
      with:
        creds: ${{ secrets.AZURE_CREDENTIALS }}
    - uses: azure/arm-deploy@v1
      with:
        resourceGroupName: github-action-arm-rg
        template: ./azuredeploy.json
        parameters: examples/template/parameters.json storageAccountType=Standard_LRS sqlS
        additionalArguments: "--what-if --rollback-on-error --what-if-exclude-change-types
```



Resource groups - Microsoft Azure

https://portal.azure.com/#view/HubsExtension/BrowseResourceGroups

Resource groups

MyAzureAD

Subscription equals **all**

Location equals **all**

0 Unsecure resources

0 Recommendations

No grouping

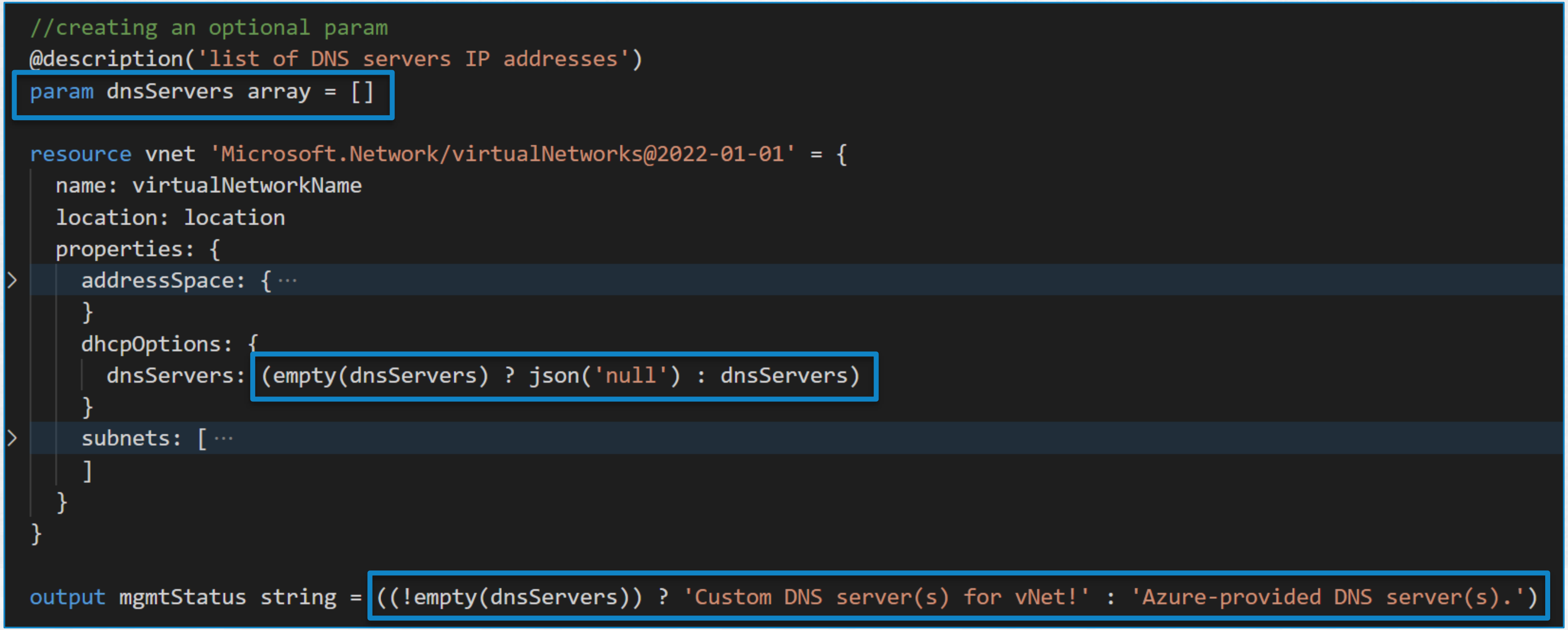
List view

Name	Subscription	Location
_rg-bicepRegistry	Visual Studio Enterprise	West Europe
_rg-ExpertsLiveDemoADO	Visual Studio Enterprise	West Europe
_rg-ExpertsLiveDemoCLI	Visual Studio Enterprise	West Europe
_rg-ExpertsLiveDemoPS	Visual Studio Enterprise	West Europe
cloud-shell-storage-westus	Visual Studio Enterprise	West US
NetworkWatcherRG	Visual Studio Enterprise	West Europe
RG-Automation	Visual Studio Enterprise	West Europe
RG-Packer	Visual Studio Enterprise	West Europe
RG-PSARM	Visual Studio Enterprise	West Europe
RGWVD	Visual Studio Enterprise	West Europe

Showing 1 to 10 of 10 records.

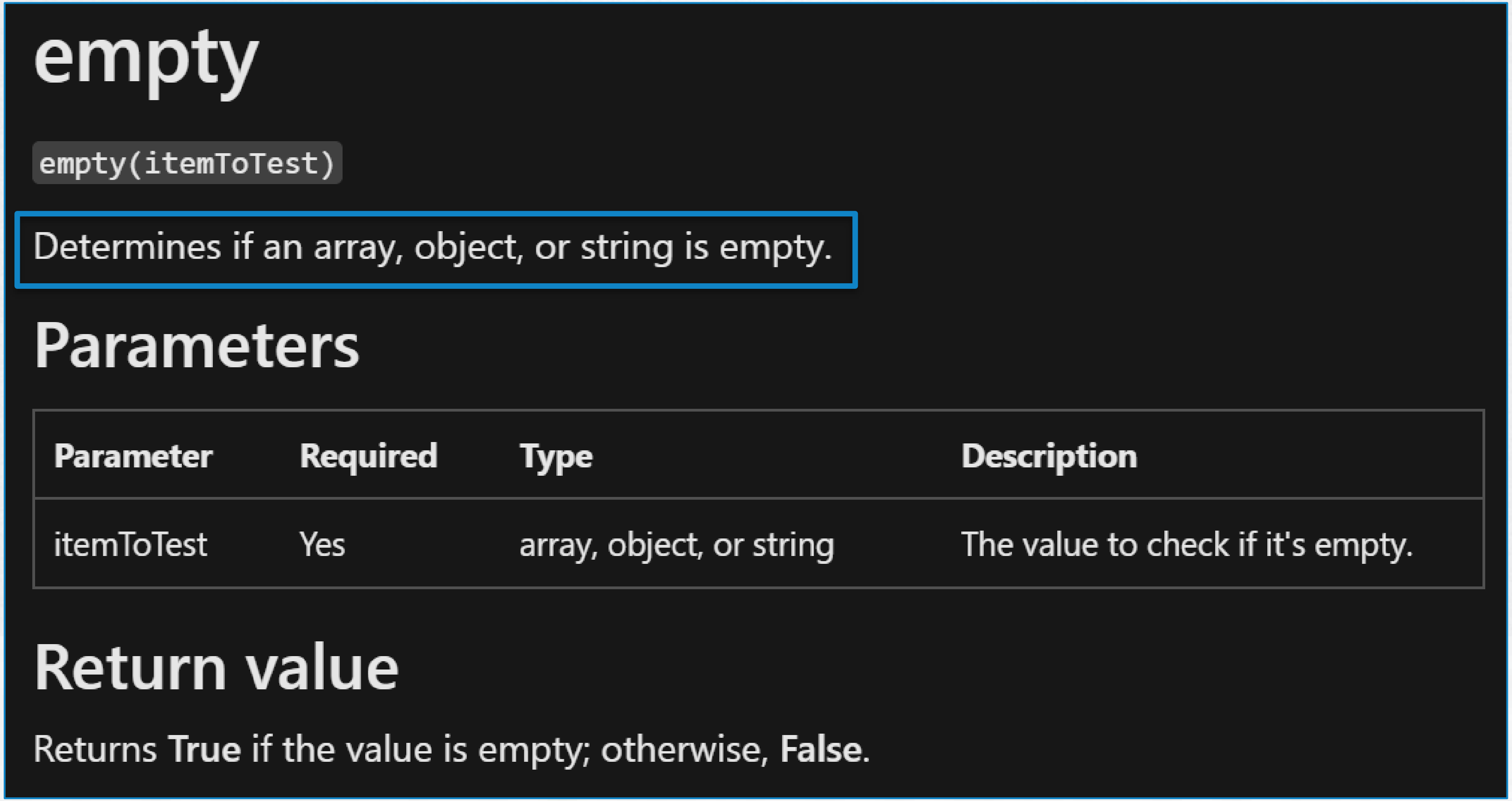
DEVOPS

Bicep tip: optional params



```
//creating an optional param
@description('list of DNS servers IP addresses')
param dnsServers array = []

resource vnet 'Microsoft.Network/virtualNetworks@2022-01-01' = {
  name: virtualNetworkName
  location: location
  properties: {
    addressSpace: {···
    }
    dhcpOptions: {
      dnsServers: (empty(dnsServers) ? json('null') : dnsServers)
    }
    subnets: [···
    ]
  }
}

output mgmtStatus string = ((!empty(dnsServers)) ? 'Custom DNS server(s) for vNet!' : 'Azure-provided DNS server(s).')
```

Bicep tip: empty()



empty

```
empty(itemToTest)
```

Determines if an array, object, or string is empty.

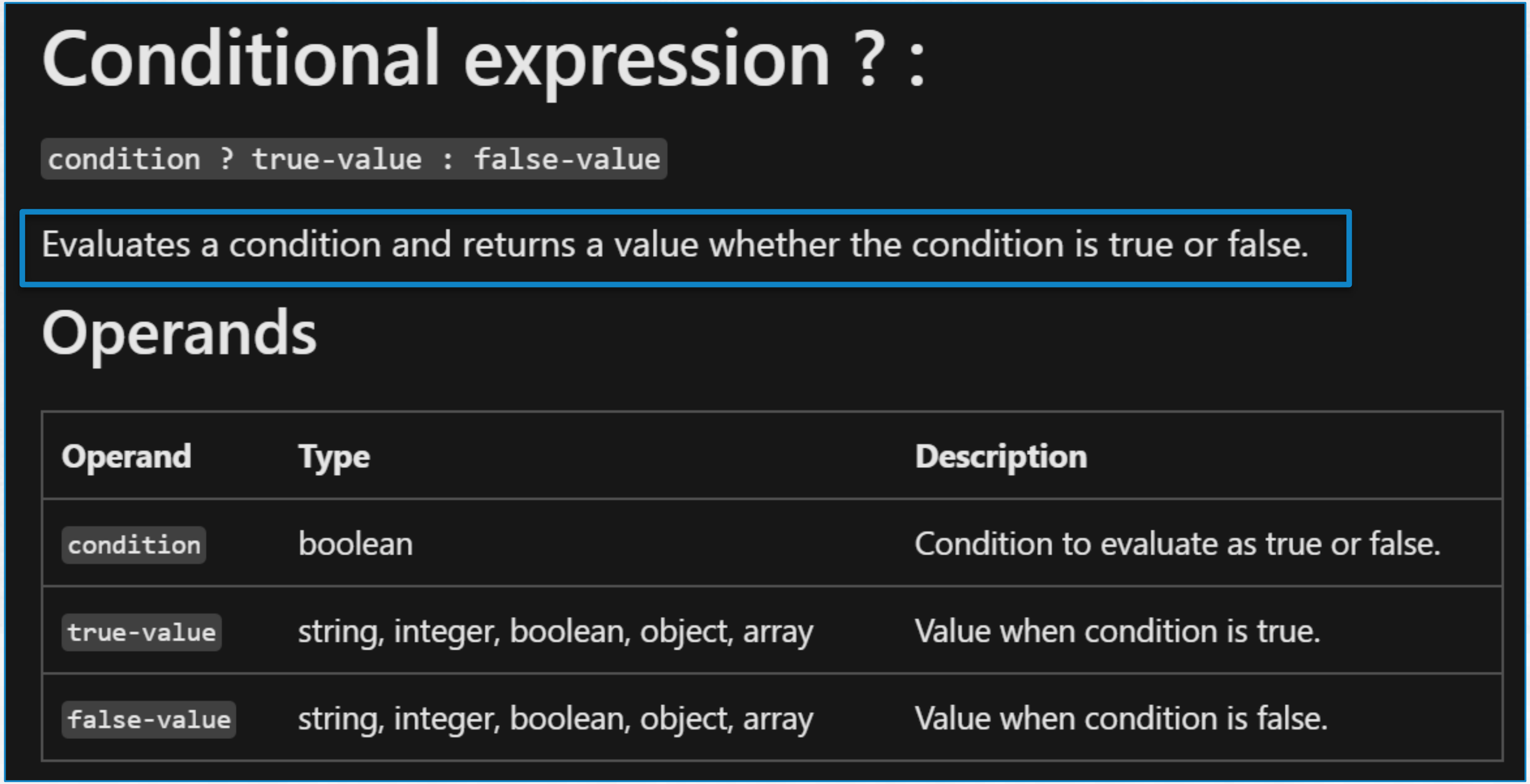
Parameters

Parameter	Required	Type	Description
itemToTest	Yes	array, object, or string	The value to check if it's empty.

Return value

Returns **True** if the value is empty; otherwise, **False**.

Bicep tip: ? :



Conditional expression ? :

```
condition ? true-value : false-value
```

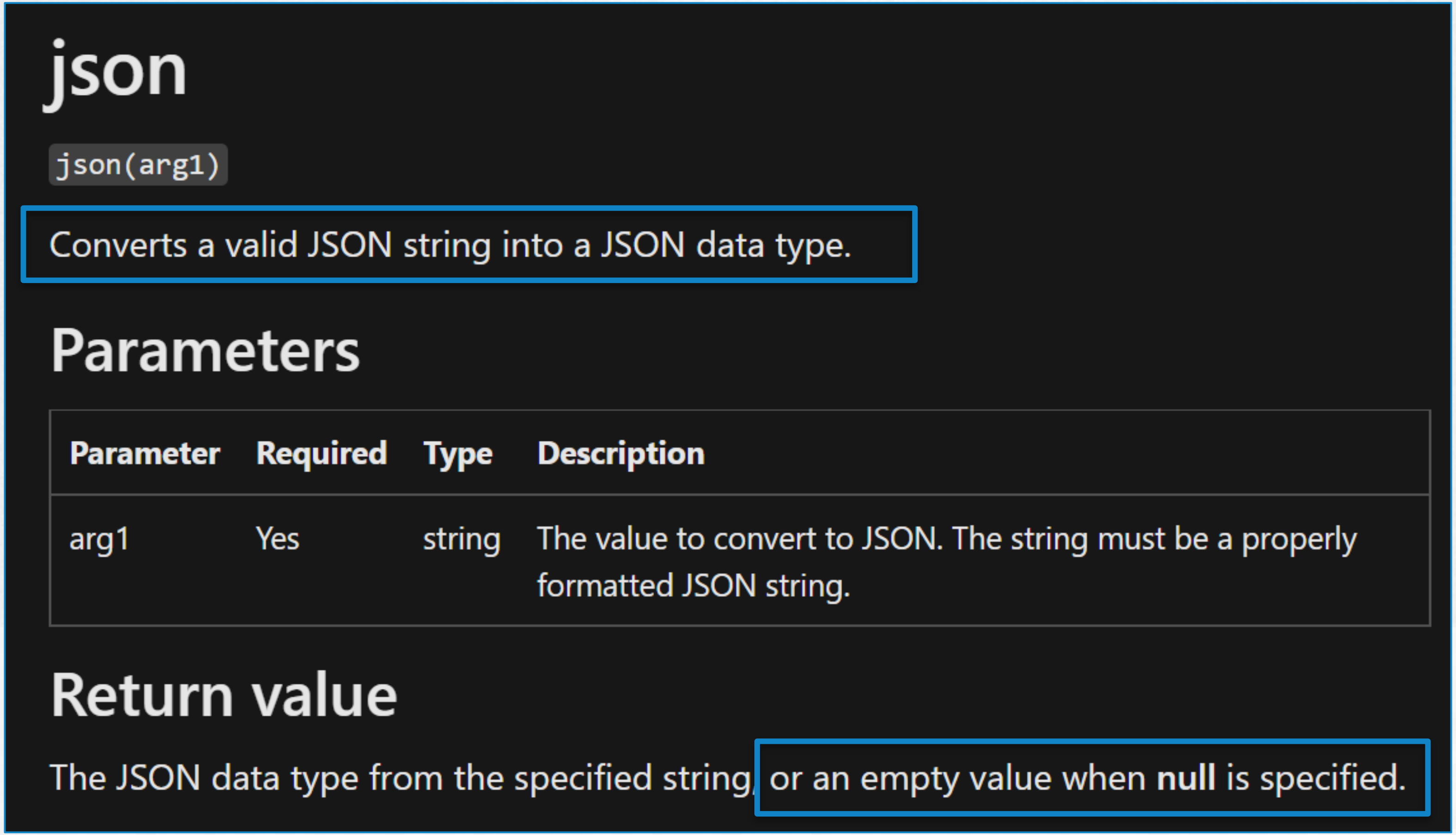
Evaluates a condition and returns a value whether the condition is true or false.

Operands

Operand	Type	Description
`condition`	boolean	Condition to evaluate as true or false.
`true-value`	string, integer, boolean, object, array	Value when condition is true.
`false-value`	string, integer, boolean, object, array	Value when condition is false.

DEVOPS

Bicep tip: json()



json

`json(arg1)`

Converts a valid JSON string into a JSON data type.

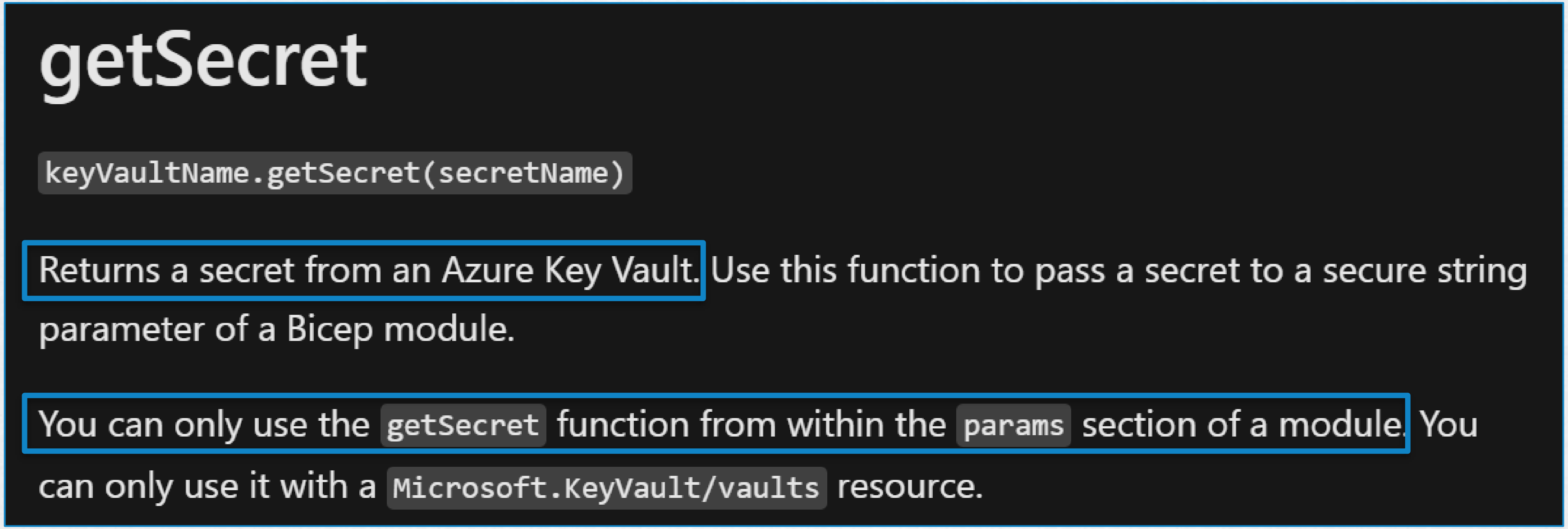
Parameters

Parameter	Required	Type	Description
arg1	Yes	string	The value to convert to JSON. The string must be a properly formatted JSON string.

Return value

The JSON data type from the specified string, or an empty value when **null** is specified.

Bicep tip: getSecret()



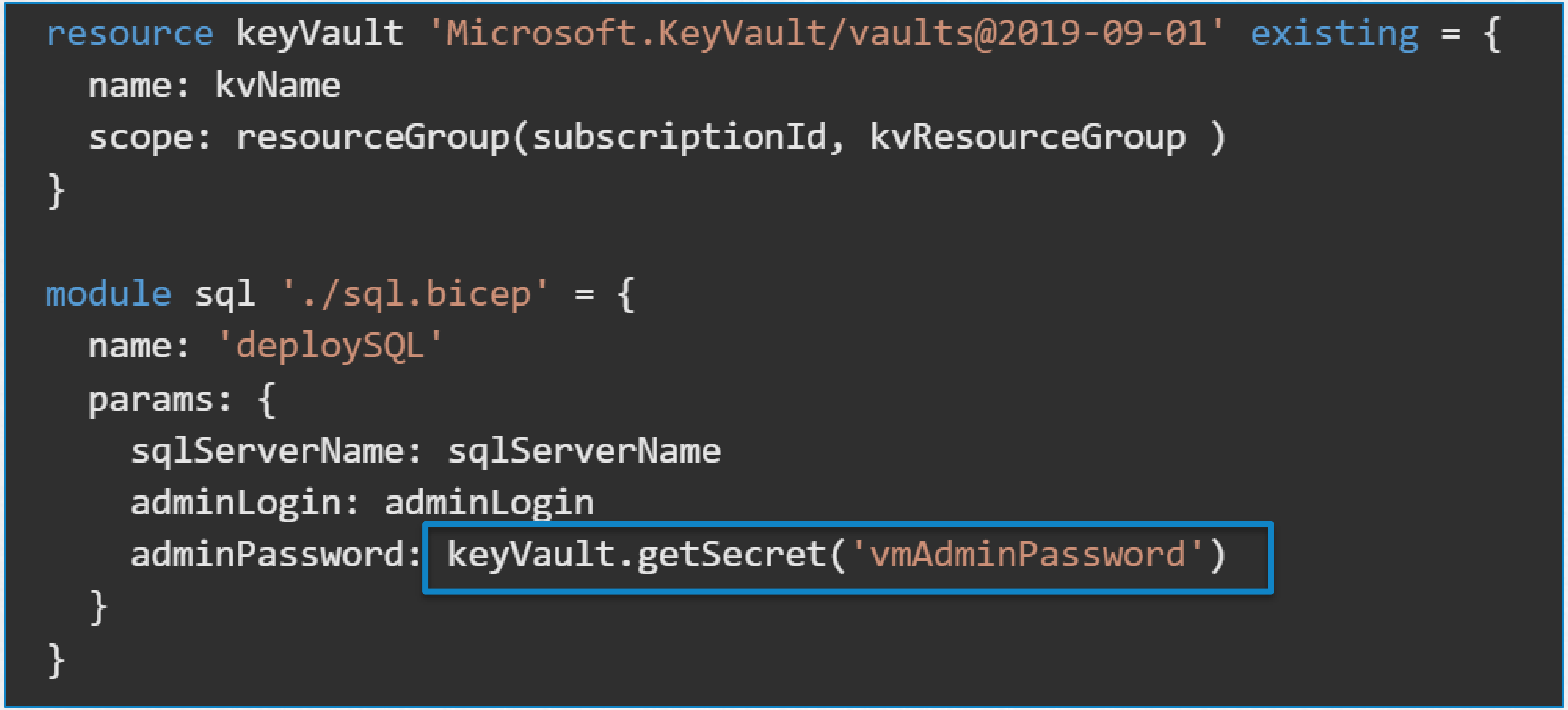
getSecret

`keyVaultName.getSecret(secretName)`

Returns a secret from an Azure Key Vault. Use this function to pass a secret to a secure string parameter of a Bicep module.

You can only use the `getSecret` function from within the `params` section of a module. You can only use it with a `Microsoft.KeyVault/vaults` resource.

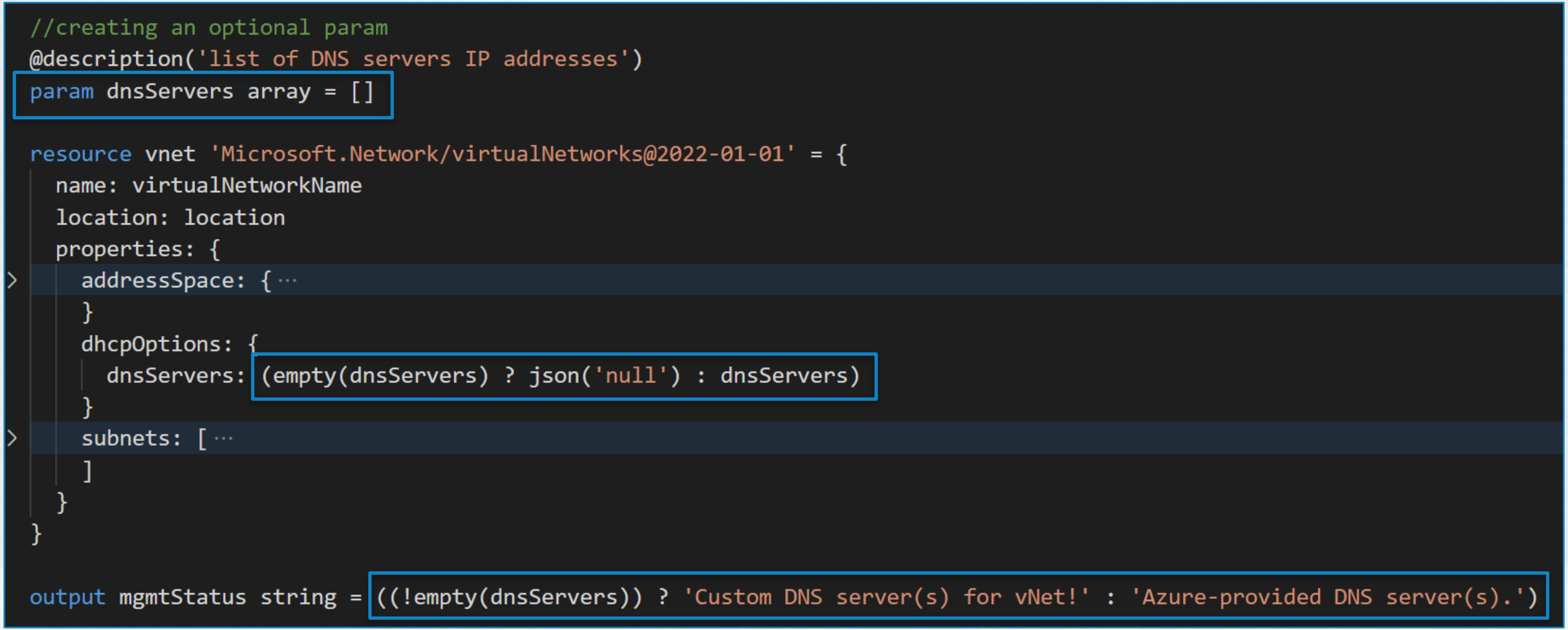
Bicep tip: getSecret()



```
resource keyVault 'Microsoft.KeyVault/vaults@2019-09-01' existing = {
  name: kvName
  scope: resourceGroup(subscriptionId, kvResourceGroup )
}

module sql './sql.bicep' = {
  name: 'deploySQL'
  params: {
    sqlServerName: sqlServerName
    adminLogin: adminLogin
    adminPassword: keyVault.getSecret('vmAdminPassword')
  }
}
```

DEVOPS

Bicep tip: json formatting

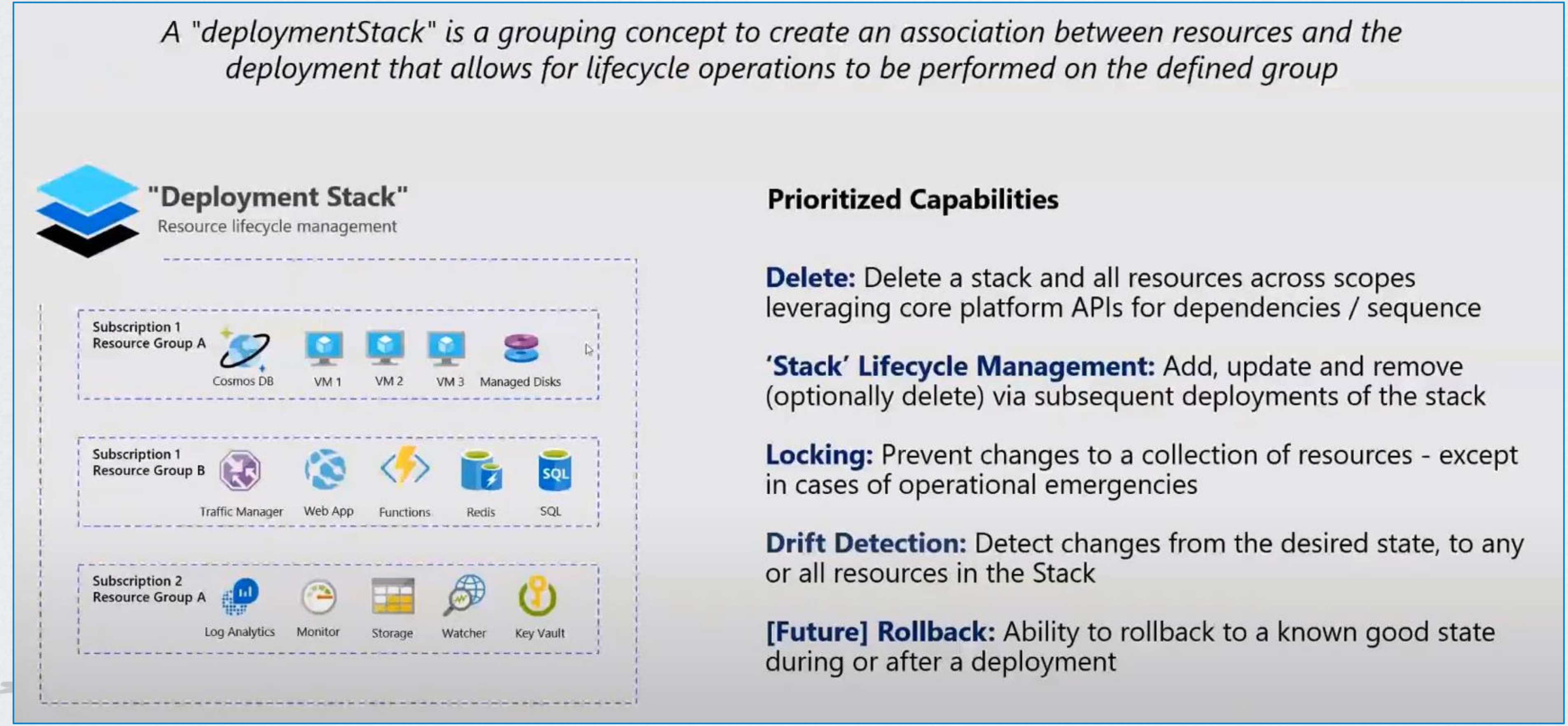


```
//creating an optional param
@description('list of DNS servers IP addresses')
param dnsServers array = []

resource vnet 'Microsoft.Network/virtualNetworks@2022-01-01' = {
  name: virtualNetworkName
  location: location
  properties: {
    addressSpace: {⋯
    }
    dhcpOptions: {
      dnsServers: (empty(dnsServers) ? json('null') : dnsServers)
    }
    subnets: [⋯
    ]
  }
}

output mgmtStatus string = ((!empty(dnsServers)) ? 'Custom DNS server(s) for vNet!' : 'Azure-provided DNS server(s).')
```

DEVOPS

Bicep tip: Deployment Stacks (preview)

https://github.com/dgfug/deployment-stacks



A "deploymentStack" is a grouping concept to create an association between resources and the deployment that allows for lifecycle operations to be performed on the defined group

"Deployment Stack"

Resource lifecycle management

Prioritized Capabilities

Delete: Delete a stack and all resources across scopes leveraging core platform APIs for dependencies / sequence

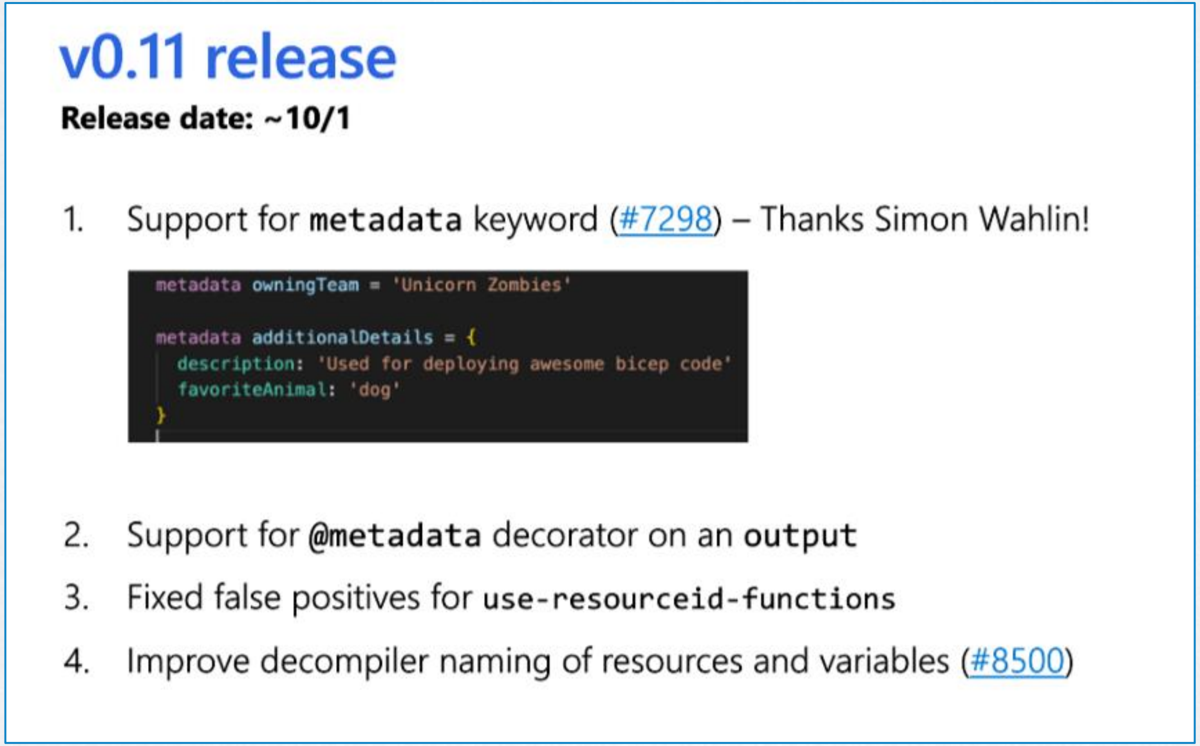
'Stack' Lifecycle Management: Add, update and remove (optionally delete) via subsequent deployments of the stack

Locking: Prevent changes to a collection of resources - except in cases of operational emergencies

Drift Detection: Detect changes from the desired state, to any or all resources in the Stack

[Future] Rollback: Ability to rollback to a known good state during or after a deployment

Bicep v0.10 – release date 10/1



v0.11 release

Release date: ~10/1

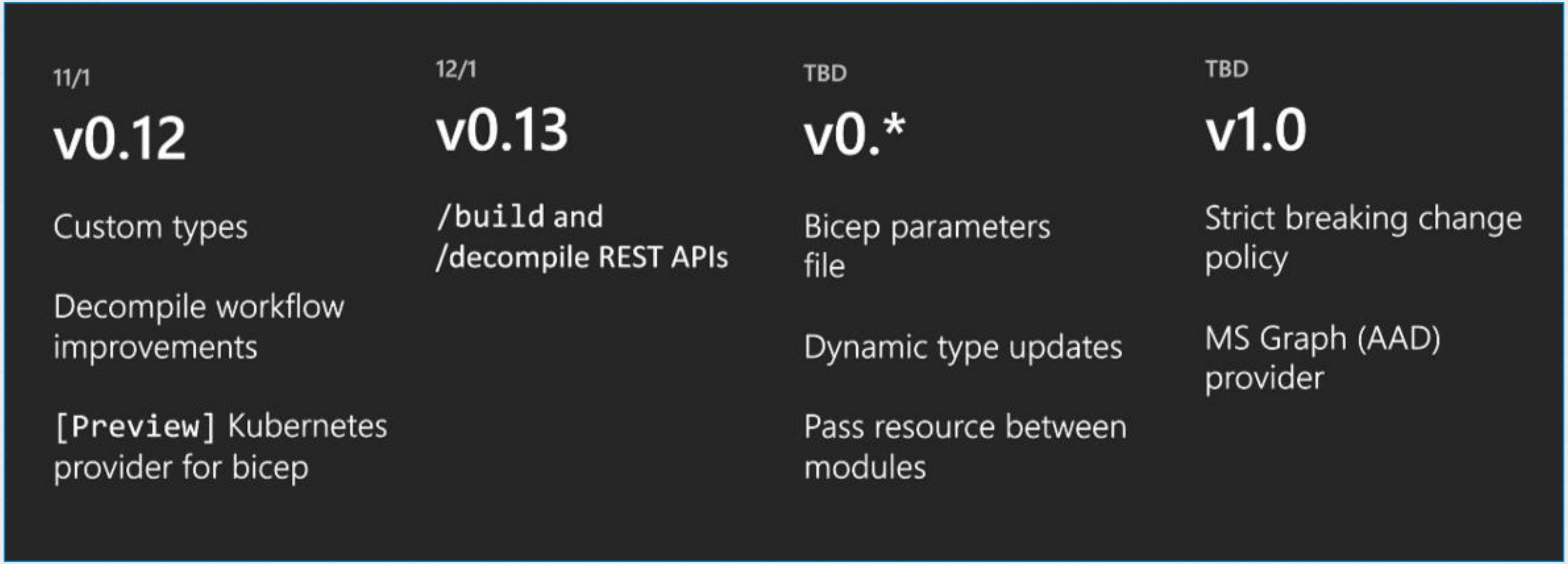
1. Support for `metadata` keyword (#7298) – Thanks Simon Wahlin!

```
metadata owningTeam = 'Unicorn Zombies'

metadata additionalDetails = {
  description: 'Used for deploying awesome bicep code'
  favoriteAnimal: 'dog'
}
```

2. Support for `@metadata` decorator on an `output`
3. Fixed false positives for `use-resourceid-functions`
4. Improve decompiler naming of resources and variables (#8500)

DEVOPS

Future Releases



11/1

v0.12

Custom types

Decompile workflow improvements

`[Preview]` Kubernetes provider for bicep

12/1

v0.13

`/build` and `/decompile` REST APIs

TBD

v0.*

Bicep parameters file

Dynamic type updates

Pass resource between modules

TBD

v1.0

Strict breaking change policy

MS Graph (AAD) provider

Delta-N Connecting the Cloud cegeka ARROW LIQUIT INSPARK Microsoft

Call to action!

- Bicep MS Docs: Aka.ms/bicep
- Bicep Monthly Community call surveymonkey.com/r/ARMnews
- Bicep GitHub location github.com/Azure/bicep
- Bicep Learning path docs.microsoft.com/en-us/azure/azure-resource-manager/bicep/learn-bicep

THANK YOU!

~~Part I: Focus on how to get started (13.00u)~~

- ~~Introducing ARM, IaC, Templates~~
- ~~Demo: Bicep & VSCode~~

~~Part II: Advanced templates & deployment (14.00)~~

- ~~Demo: Advanced Bicep Language capabilities~~
- ~~Demo: Deployment methods~~