# Infrastructure as Code with Bicep Road show!


*Esther Barthel*

*@virtuEs_IT*

*github.com/cognitionit*

*Microsoft MVP*



*Freek Berson*

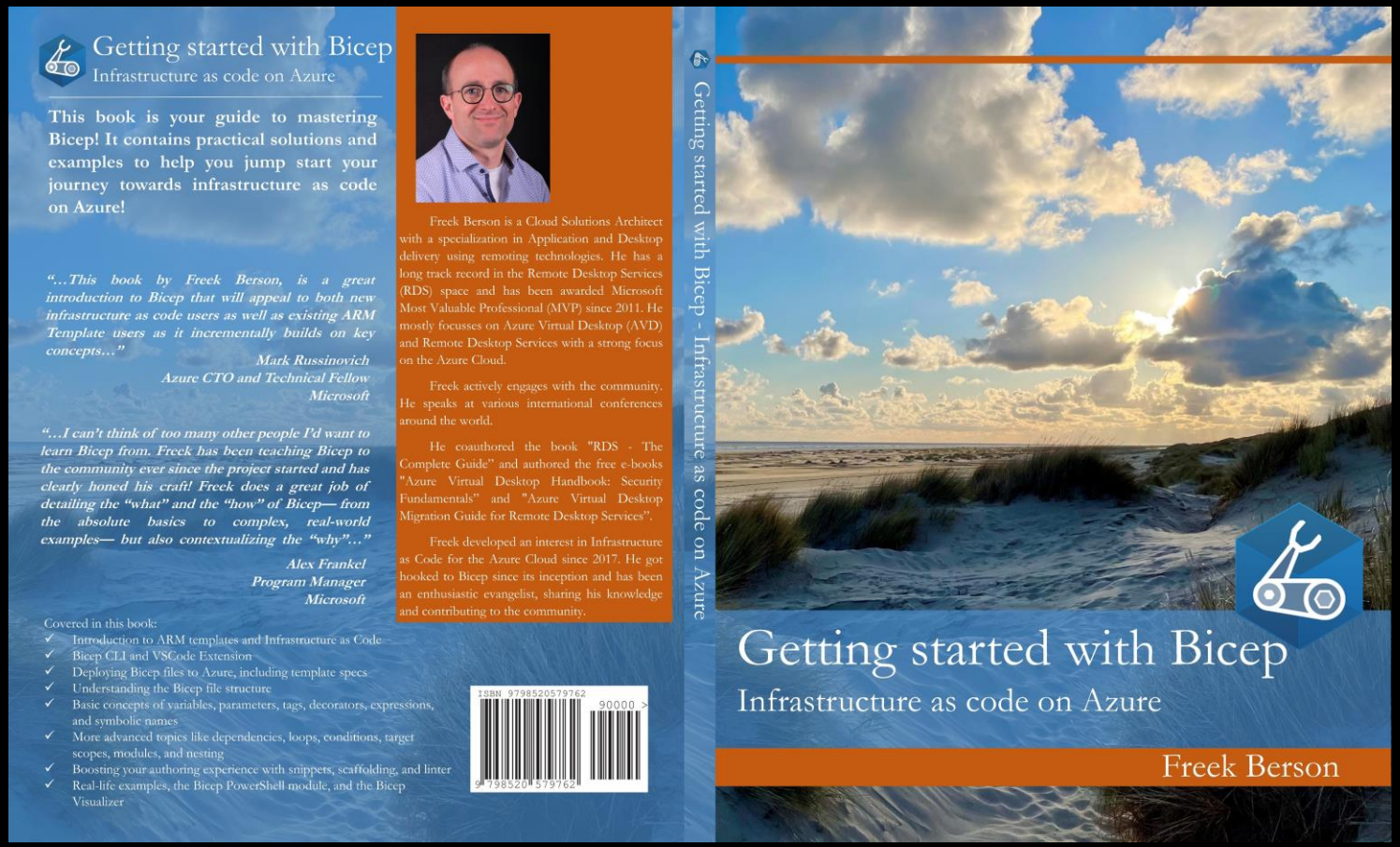
*@fberson*

*github.com/fberson*

*Microsoft MVP*


*Create a Tweet or LinkedIn post about this session and win a copy of the Getting started with Bicep book!*

*@fberson*
*@virtuEs_IT*



# Infrastructure as Code with Bicep

Roadshow

2021

| | |
|---|---|
| January | Azure Lowlands |
| February | Azure Thursday |
| February | Ask Wortell Live Stream |
| April | Dutch DevOps Community |
| April | Detron knowledge event |
| April | Technine User Group |
| May | Belgium AVD User Group |
| June | Azure Meetup Oslo |
| July | US AVD User Group |
| August | Azure User Group Norway |

2022

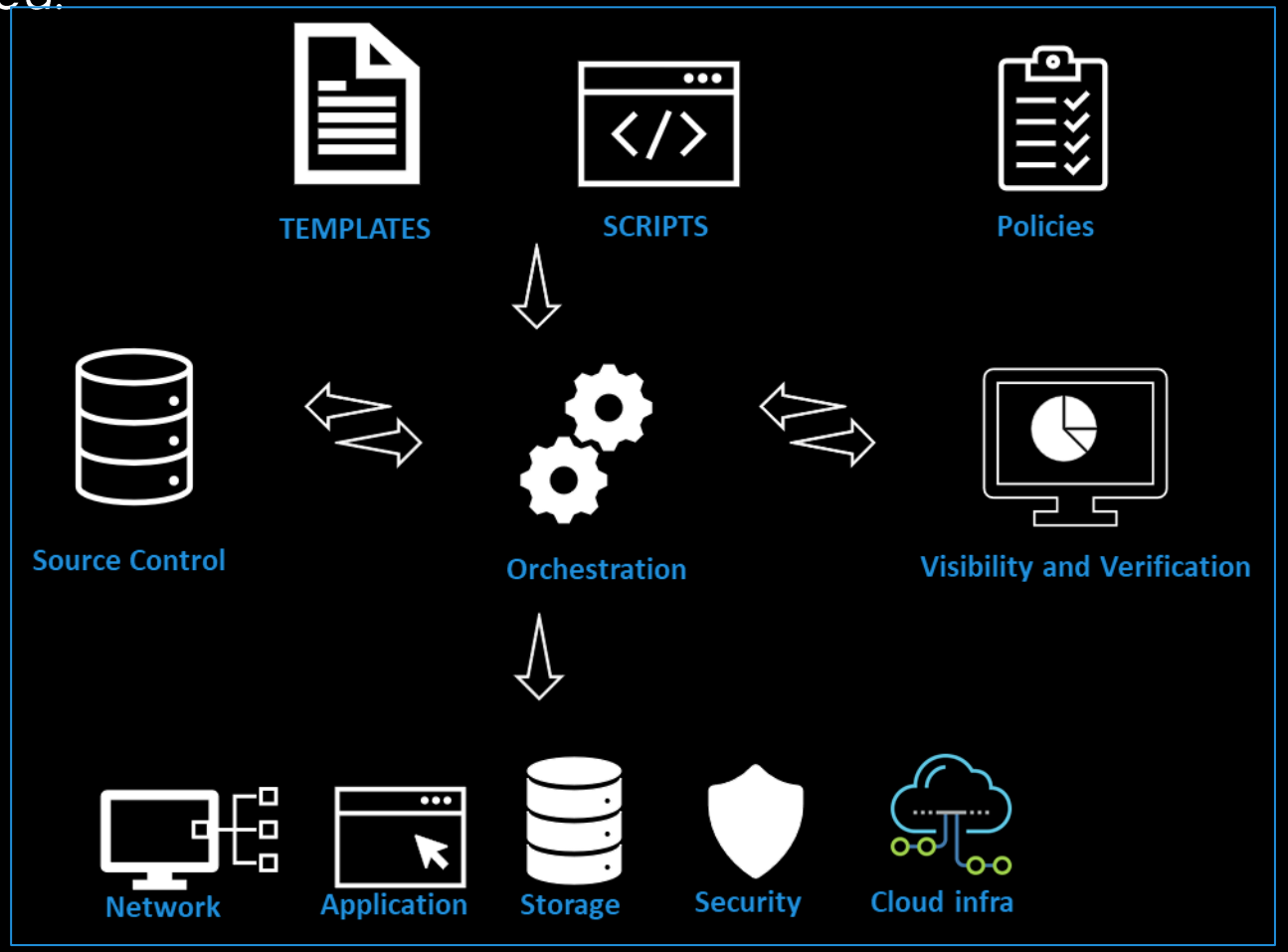
| | |
|---|---|
| February | Azure User Group Iceland |
| March | MC2MC Belgium |
| April | AVD Tech Fest, Amsterdam |
| June | Scottish Summit, Glasgow |

@virtuEs_IT

@fberson

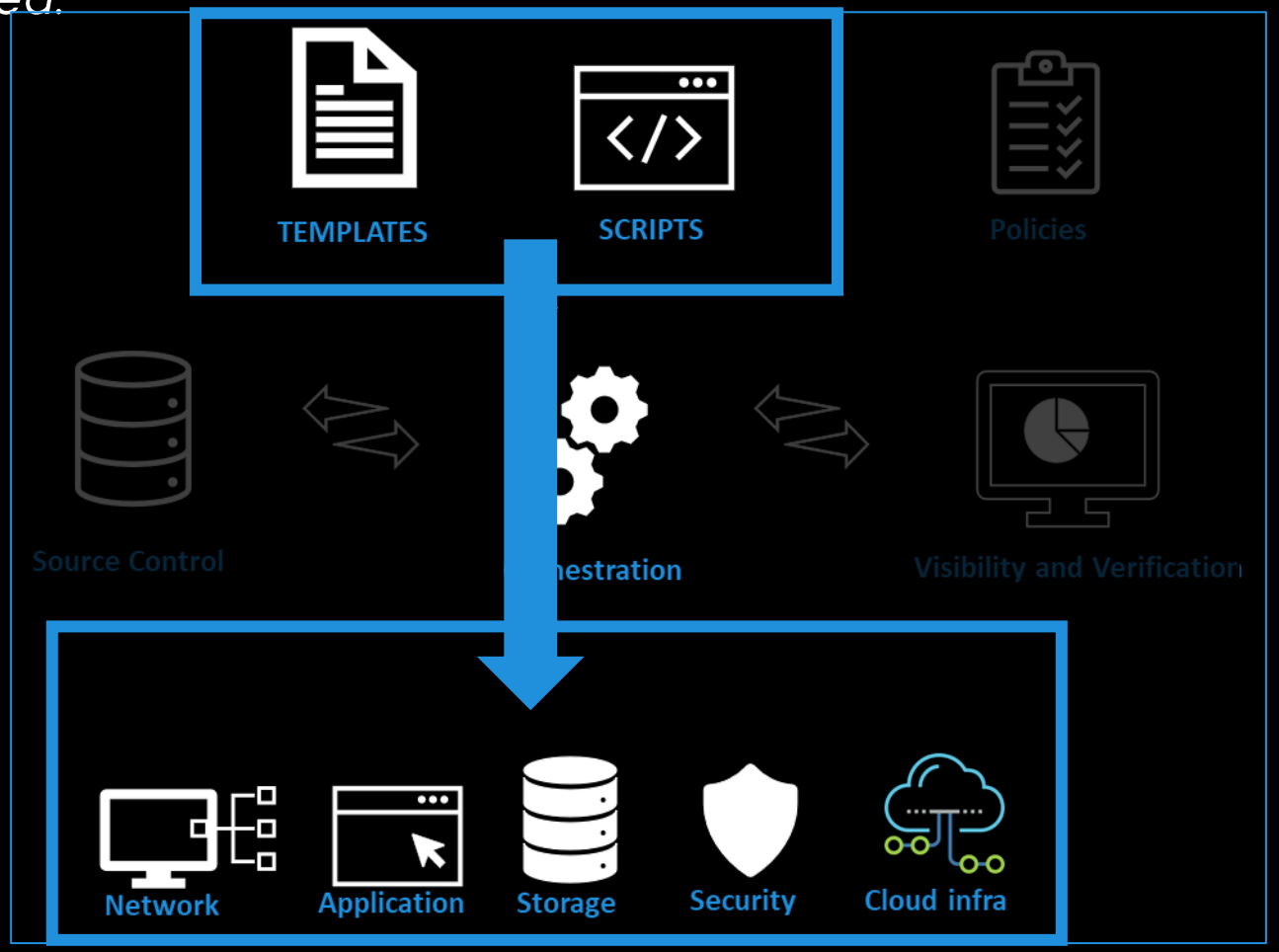
# Infrastructure as Code (IaC)

*"..the process of provisioning infrastructure resources similar to how software is deployed."*

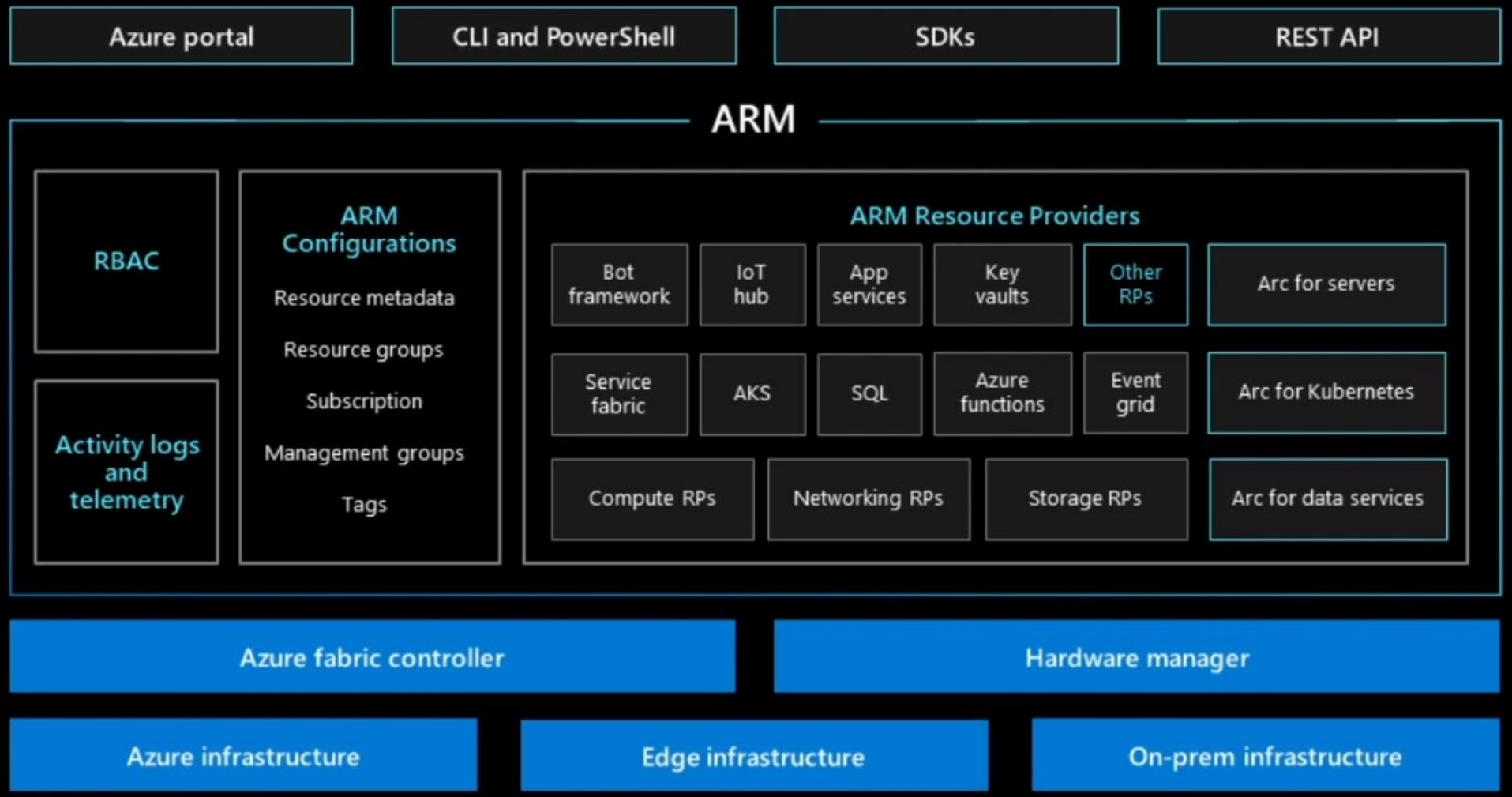


# Infrastructure as Code (IaC)

*"..the process of provisioning infrastructure resources similar to how software is deployed."*

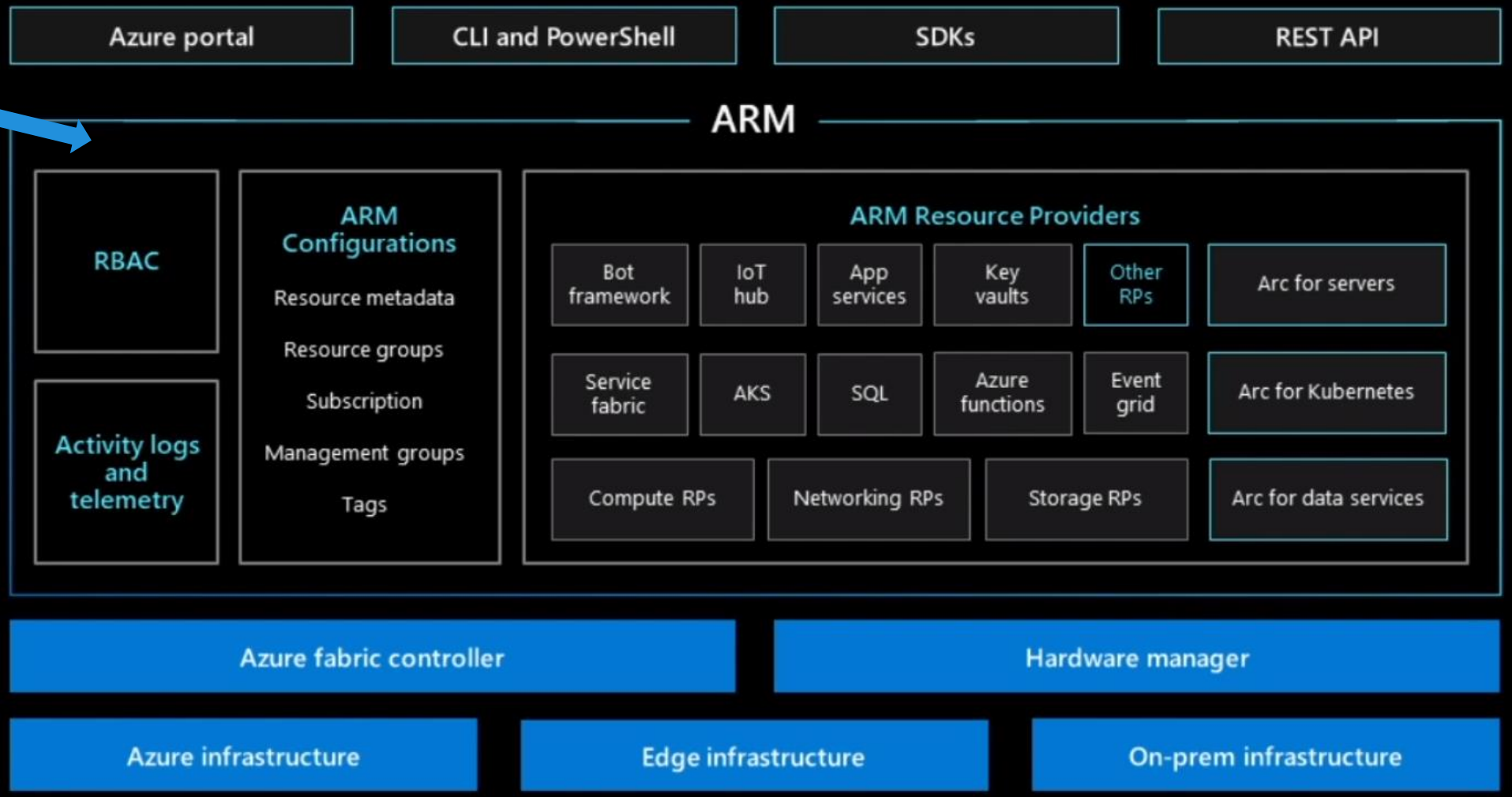


# Azure Resource Manager



# Azure Resource Manager

ARM Template



# Infrastructure as code

## Template format

In its simplest structure, a template has the following elements:

JSON Copy

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "",
  "apiProfile": "",
  "parameters": {  },
  "variables": {  },
  "functions": [  ],
  "resources": [  ],
  "outputs": {  }
}
```

# ARM Template complexity

```
605                   "hostPoolArmPath": "[resourceId('Microsoft.DesktopVirtualization/hostPools', format('{0}-REMOTEAPP', parameters('hostpoolName')))]"
606                 },
607                 "dependsOn": [
608                   "[resourceId('Microsoft.DesktopVirtualization/hostPools', format('{0}-REMOTEAPP', parameters('hostpoolName')))]"
609                 ]
610               },
611               {
612                 "type": "Microsoft.DesktopVirtualization/workspaces",
613                 "apiVersion": "2019-12-10-preview",
614                 "name": "[parameters('workspaceName')]",
615                 "location": "[parameters('AVDbackplanelocation')]",
616                 "properties": {
617                   "friendlyName": "[parameters('workspaceNameFriendlyName')]",
618                   "applicationGroupReferences": [
619                     "[resourceId('Microsoft.DesktopVirtualization/applicationGroups', parameters('appgroupName'))]",
620                     "[if(parameters('createRemoteAppHostpool'), resourceId('Microsoft.DesktopVirtualization/applicationGroups', format('{0}-REMOTEAPP', parameters('appgroupName'))), '')]"
621                   ]
622                 },
623                 "dependsOn": [
624                   "[resourceId('Microsoft.DesktopVirtualization/applicationGroups', parameters('appgroupName'))]",
625                   "[resourceId('Microsoft.DesktopVirtualization/applicationGroups', format('{0}-REMOTEAPP', parameters('appgroupName')))]"
626                 ]
627               }
628             ]
629           }
630         },
631         "dependsOn": [
632           "[subscriptionResourceId('Microsoft.Resources/resourceGroups', format('{0}BACKPLANE{1}', parameters('resourceGroupProdPrefix'), parameters('resourceGroupPostfix')))]"
633         ]
634       },
635       {
636         "type": "Microsoft.Resources/deployments",
637         "apiVersion": "2020-06-01",
```

# Project ‘Bicep’

# 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?

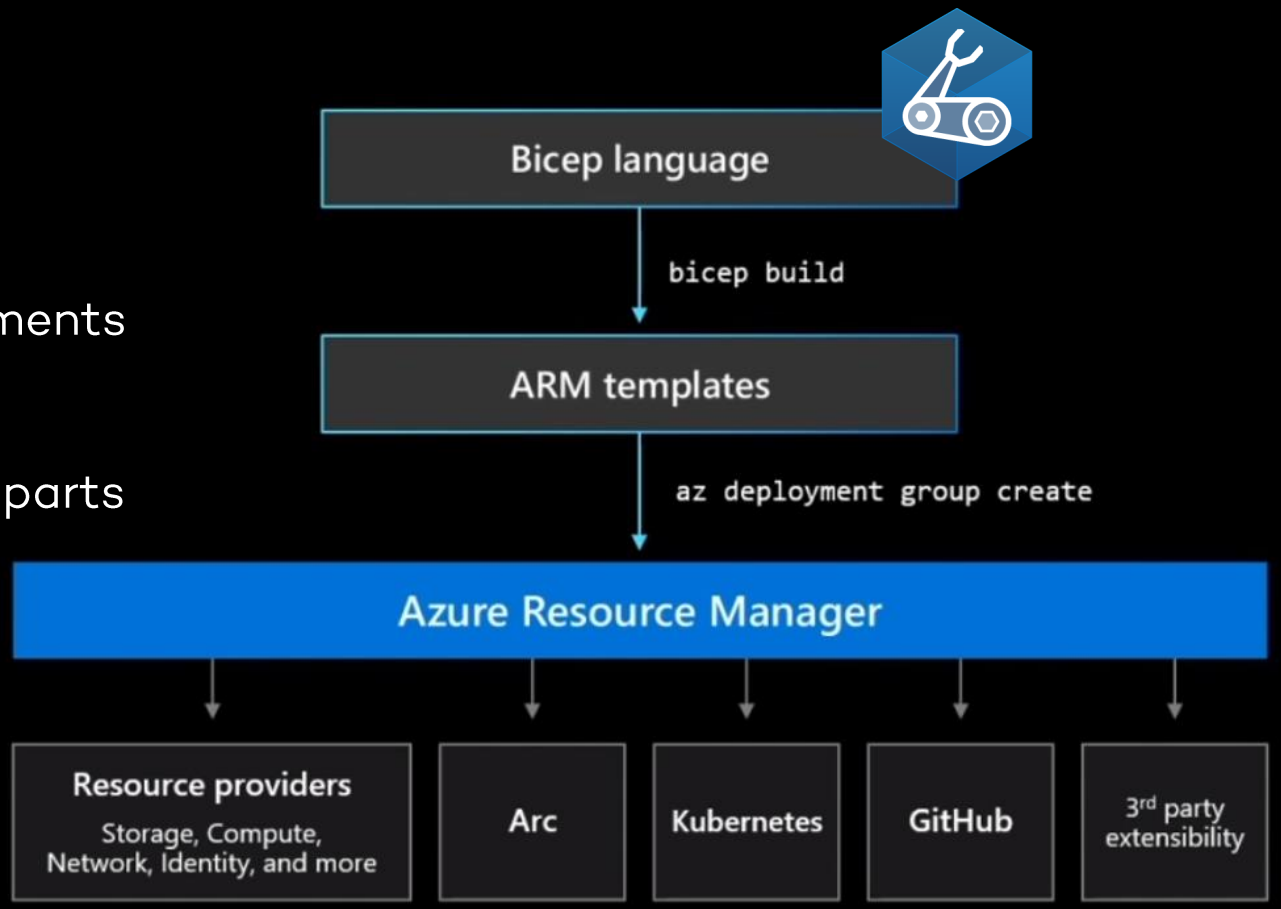
*Simple declarative language to provision infrastructure to Azure.*

*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

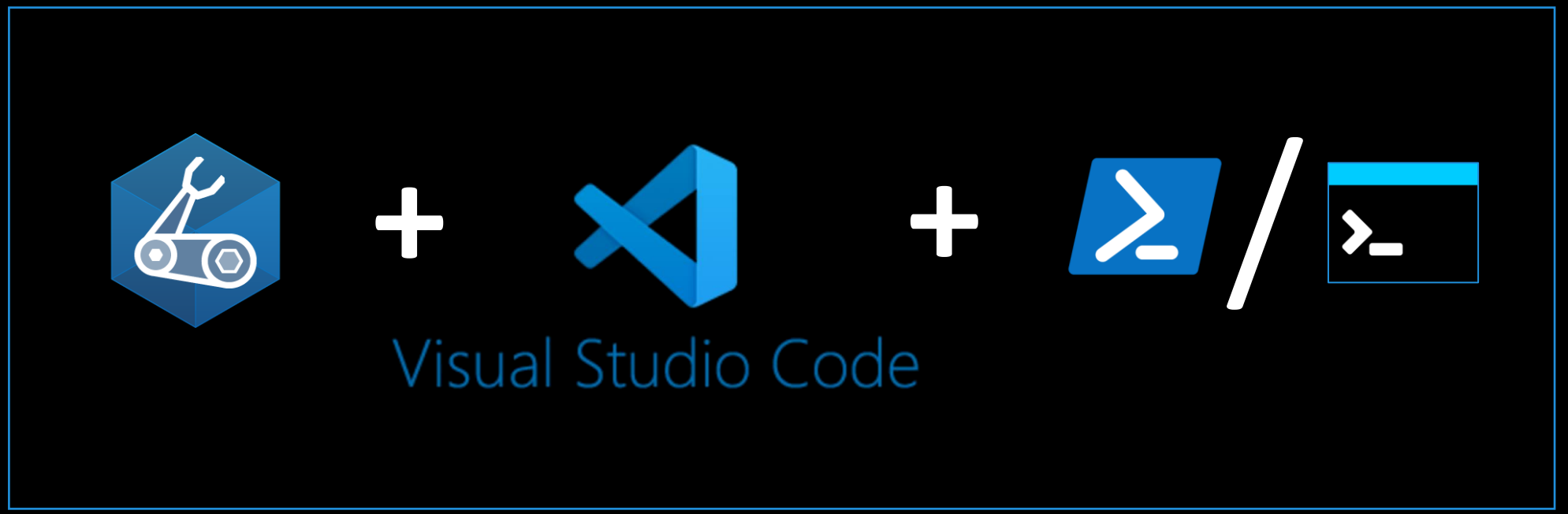
Open Source
Transparency and community



# Get started with Bicep

(https://aka.ms/bicep)

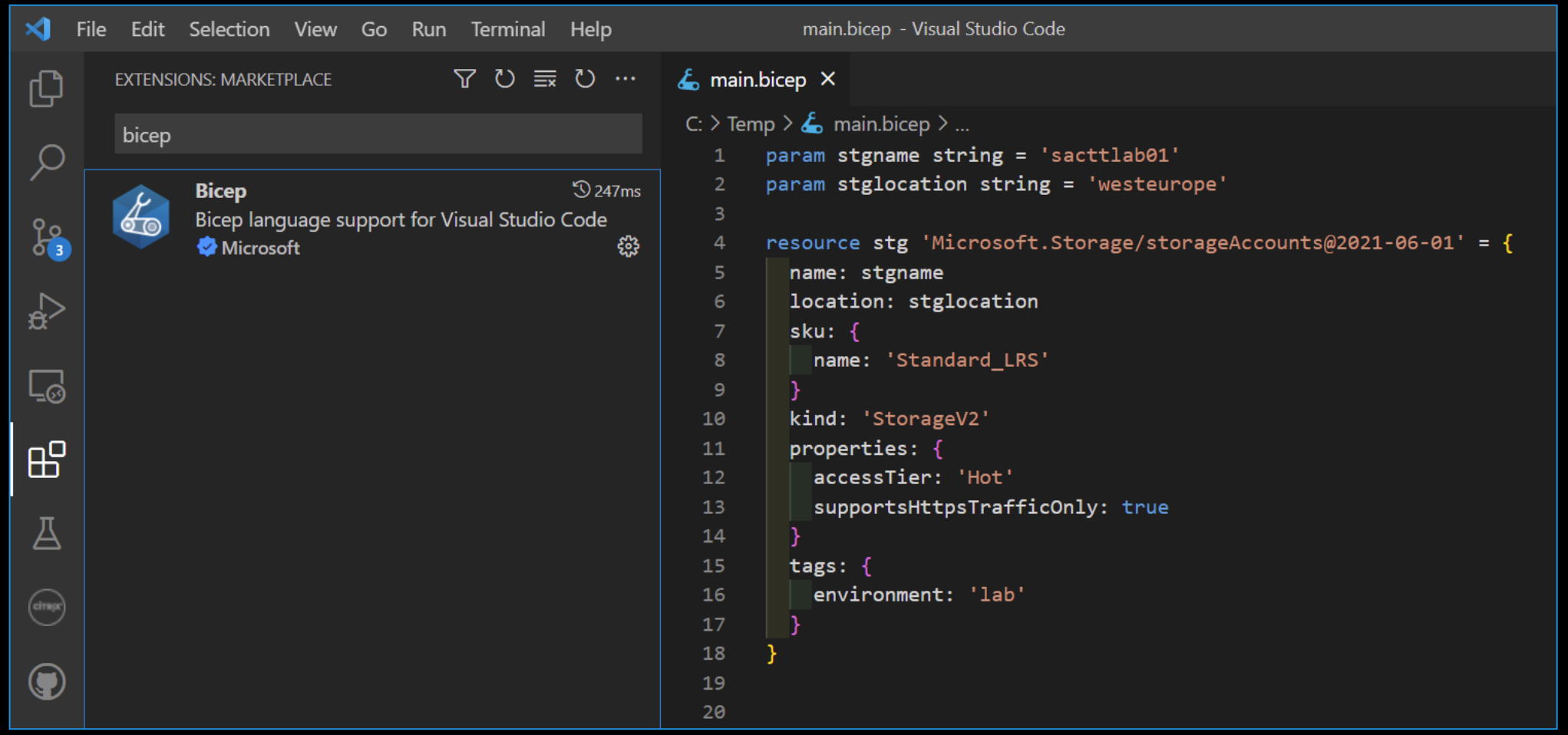
1. Install the tools (bicep.exe, Visual Studio Code, PowerShell/az cli)



# Get started with Bicep

(https://aka.ms/bicep)

## 2. Create a Bicep template



# Get started with Bicep

(https://aka.ms/bicep)

## 3. Deploy the resources in Azure (using PowerShell/az cli)

DeployBicepTemplate.ps1 1

C: > Temp > DeployBicepTemplate.ps1 > ...

```powershell
# Authenticate into your Azure subscription
$azSession = Connect-AzAccount

# create a parameter object for the template input parameters
$templateParameterInput =@{
    stgname = 'sactt08'
    stglocation = 'westeurope'
}

# Create parameter object for deployment
$cmdletParamInput = @{
    "ResourceGroupName" = "rg-lab-storage"
    "TemplateFile" = "main.bicep"
    "TemplateParameterObject" = $templateParameterInput
}

# Deploy the resource group with a bicep template directly
New-AzResourceGroupDeployment @cmdletParamInput
```

# Demo!

# Road map

v0.1

(aug '20)

Alpha Release available on August 31st

v0.2

(Oct '20)

- VSCode Intellisense
- Support for modules

v0.3

(March '21)

- Loops
- Conditionals
- Decompiler
- Production usage

v0.4

- Quality release
- Learn module
- Linter (TTK successor)
- Snippets & resource scaffolding
- Merging ARM Quickstarts & bicep
- IncludeFile() support

NOV '21

v0.5

- Public & private Module Registry
- Linter vNext
- Passing resource to module
- LoadTextContent(…)

March 2022

V1.0

- Strict change policy
- Separate type & core updates
- Bicep Extensibility preview

# Infrastructure as Code with Bicep Road show!

# Thank you!


***Esther Barthel***

***@virtuEs_IT***

***github.com/cognitionit***

***Microsoft MVP***



***Freek Berson***

***@fberson***

***github.com/fberson***

***Microsoft MVP***