Bicep – Infrastructure as Code Masterclass – part I

Esther Barthel - @virtuEs_IT

Freek Berson - @fberson

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/Azure since 2017

Freek Berson

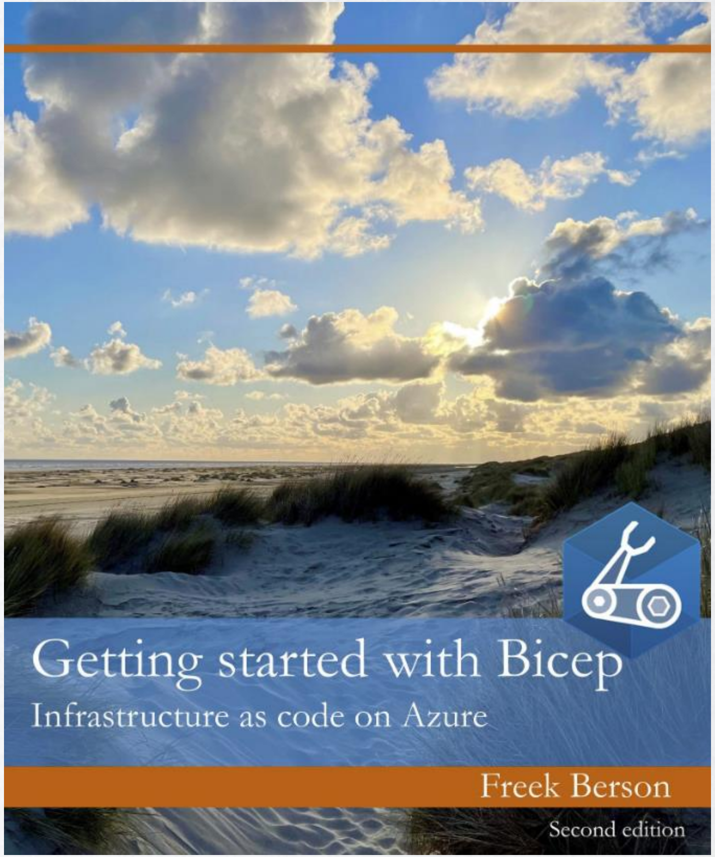
Principal Outbound Product Manager
Parallels (Alludo)
@fberson

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

DEVOPS

Tweet to win!

@fberson
@virtuEs_IT
#ExpertsLiveNL



Delta-N Connecting the Cloud · cegeka · ARROW · LIQUIT · INSPARK · Microsoft

Agenda

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

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

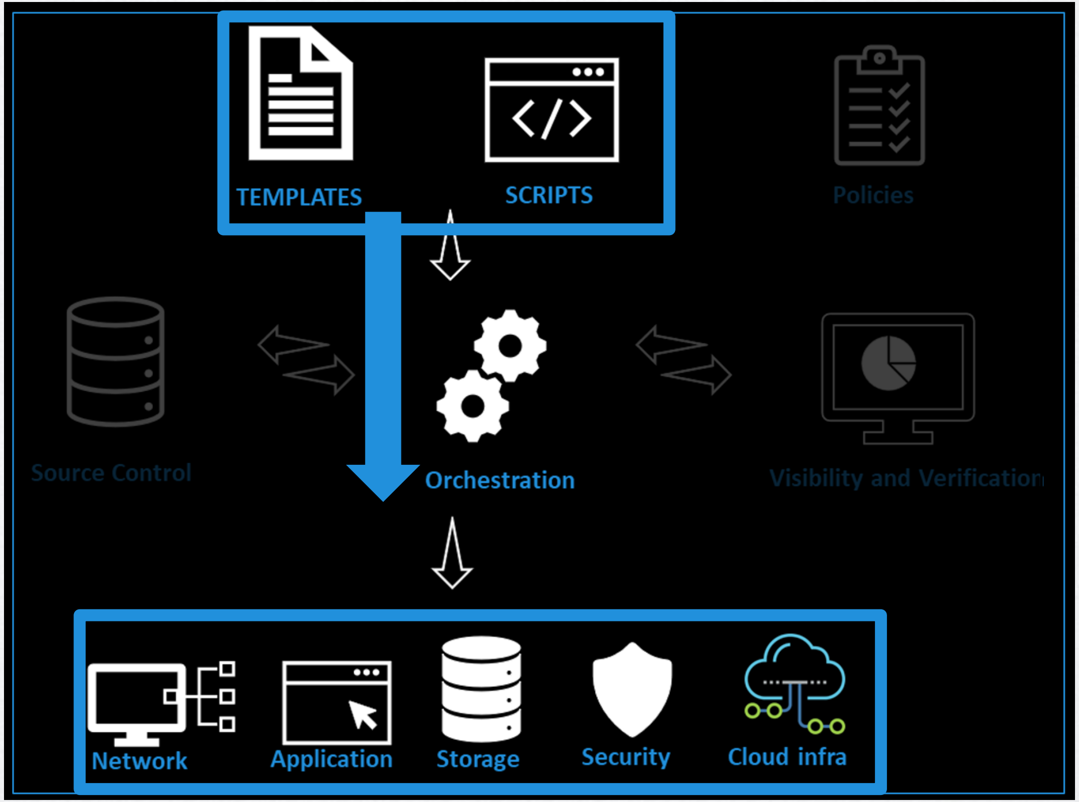
Part II: Advanced templates & deployment (14.00)

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

DEVOPS

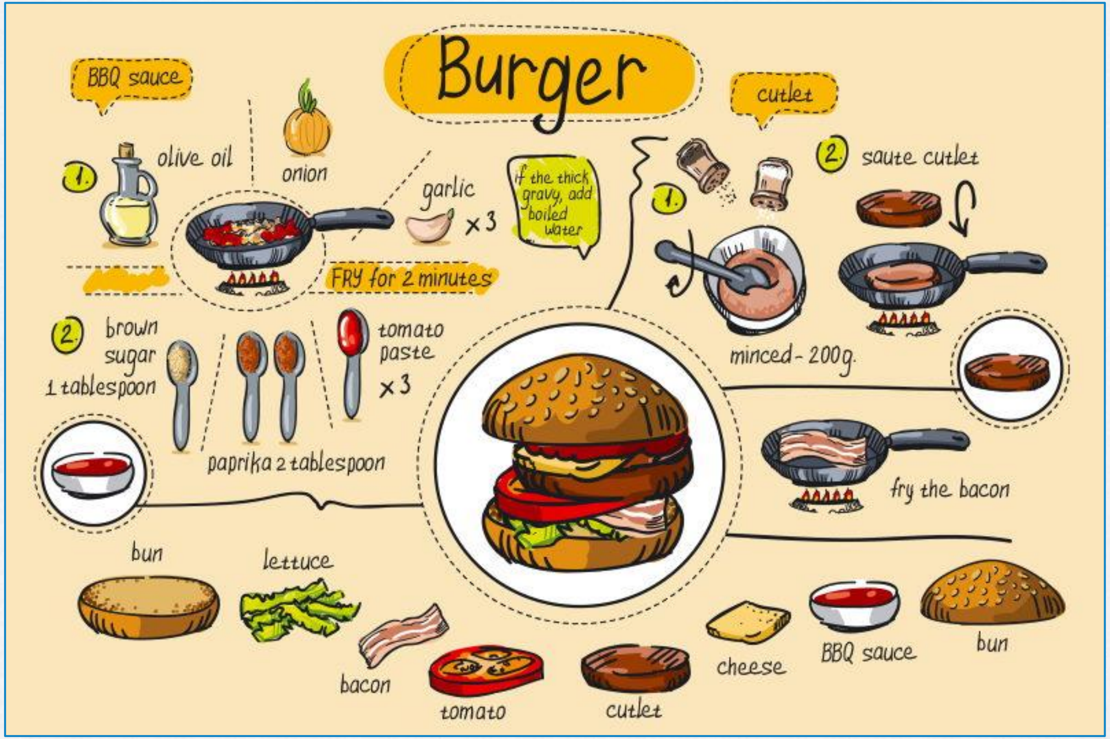
Infrastructure as Code (IaC)

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



Imperative code

You **execute a sequence of commands**, in a specific order, to reach an end configuration. This process defines what the code should accomplish, and it defines how to accomplish the task like a **step-by-step instruction** manual

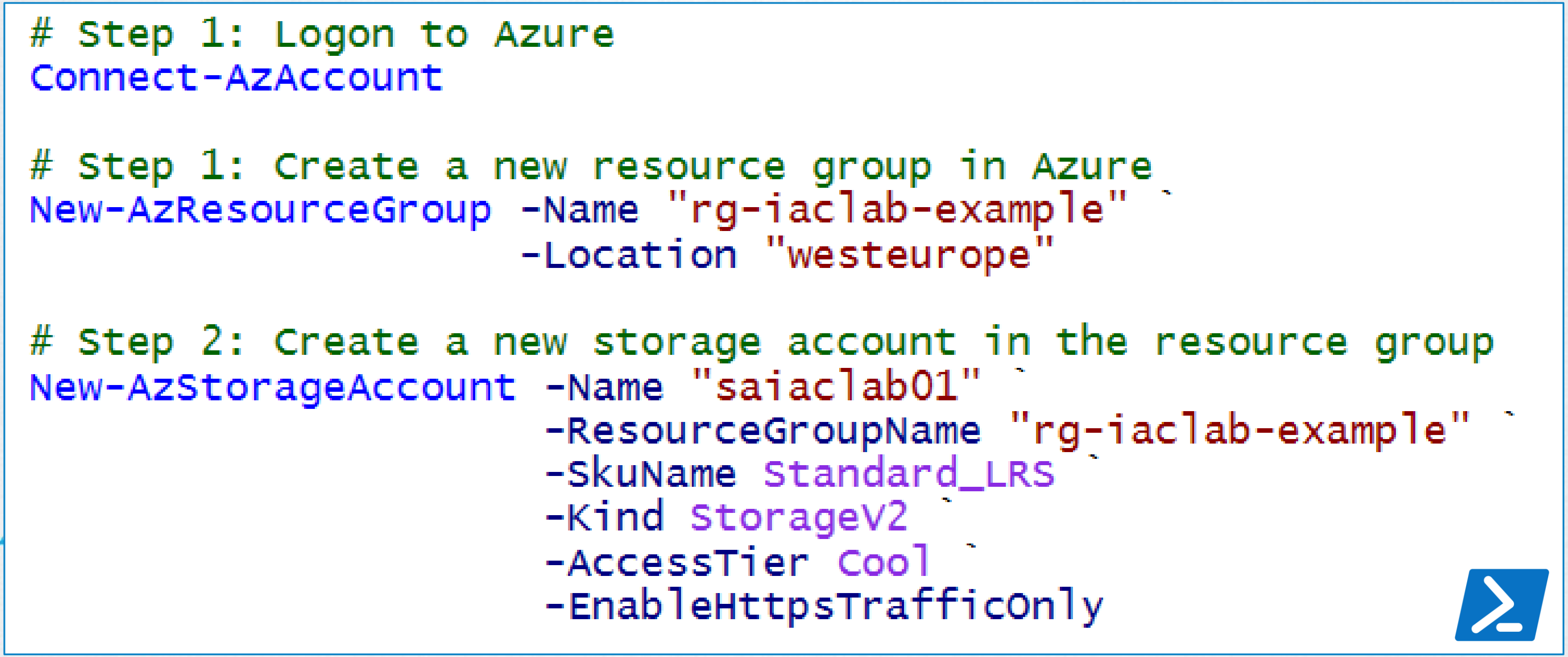


DEVOPS

Imperative code

You **execute a sequence of commands**, in a specific order, to reach an end configuration.

Using **scripts**.



```
# Step 1: Logon to Azure
Connect-AzAccount

# Step 1: Create a new resource group in Azure
New-AzResourceGroup -Name "rg-iaclab-example" `
                    -Location "westeurope"

# Step 2: Create a new storage account in the resource group
New-AzStorageAccount -Name "saiaclab01" `
                     -ResourceGroupName "rg-iaclab-example" `
                     -SkuName Standard_LRS `
                     -Kind StorageV2 `
                     -AccessTier Cool `
                     -EnableHttpsTrafficOnly
```

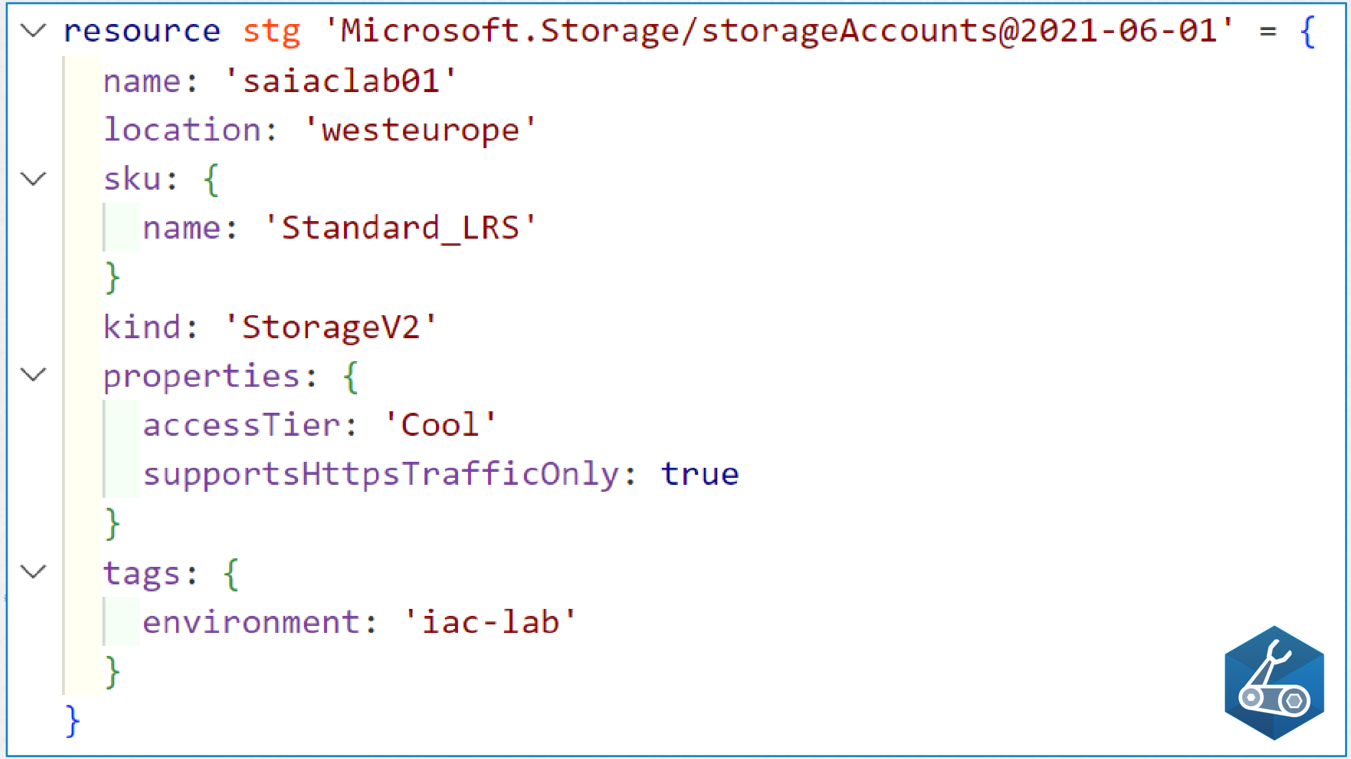
Declarative code

You **specify only the end configuration**. The code doesn't define how to accomplish the task. Like **ordering from a menu**.

Declarative code

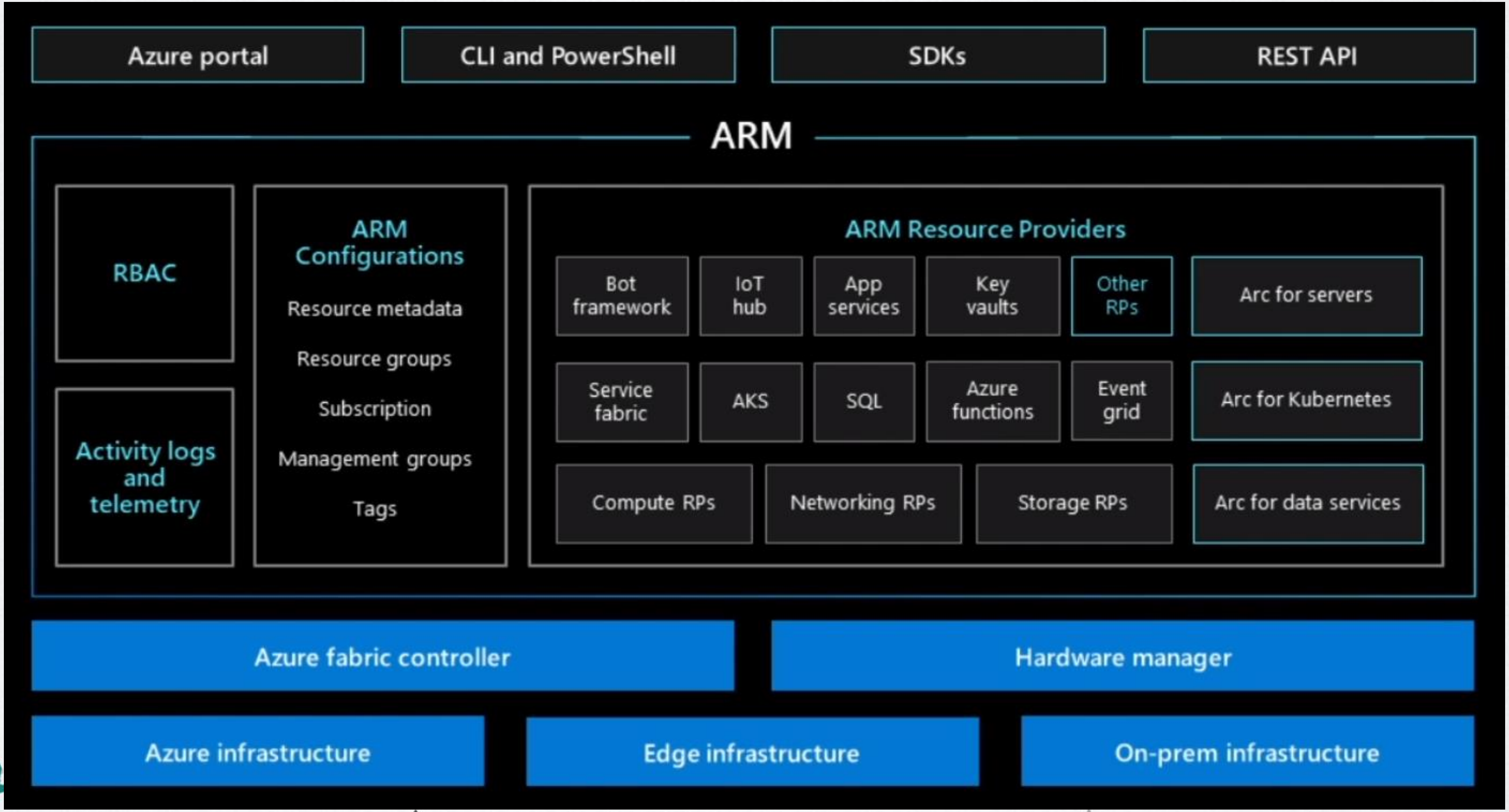
You **specify only the end configuration**. The code doesn't define how to accomplish the task.

Using **templates**.



```
resource stg 'Microsoft.Storage/storageAccounts@2021-06-01' = {
  name: 'saiaclab01'
  location: 'westeurope'
  sku: {
    name: 'Standard_LRS'
  }
  kind: 'StorageV2'
  properties: {
    accessTier: 'Cool'
    supportsHttpsTrafficOnly: true
  }
  tags: {
    environment: 'iac-lab'
  }
}
```

Azure Resource Manager



Azure portal

CLI and PowerShell

SDKs

REST API

ARM

RBAC

Activity logs and telemetry

ARM Configurations

Resource metadata

Resource groups

Subscription

Management groups

Tags

ARM Resource Providers

Bot framework

IoT hub

App services

Key vaults

Other RPs

Arc for servers

Service fabric

AKS

SQL

Azure functions

Event grid

Arc for Kubernetes

Compute RPs

Networking RPs

Storage RPs

Arc for data services

Azure fabric controller

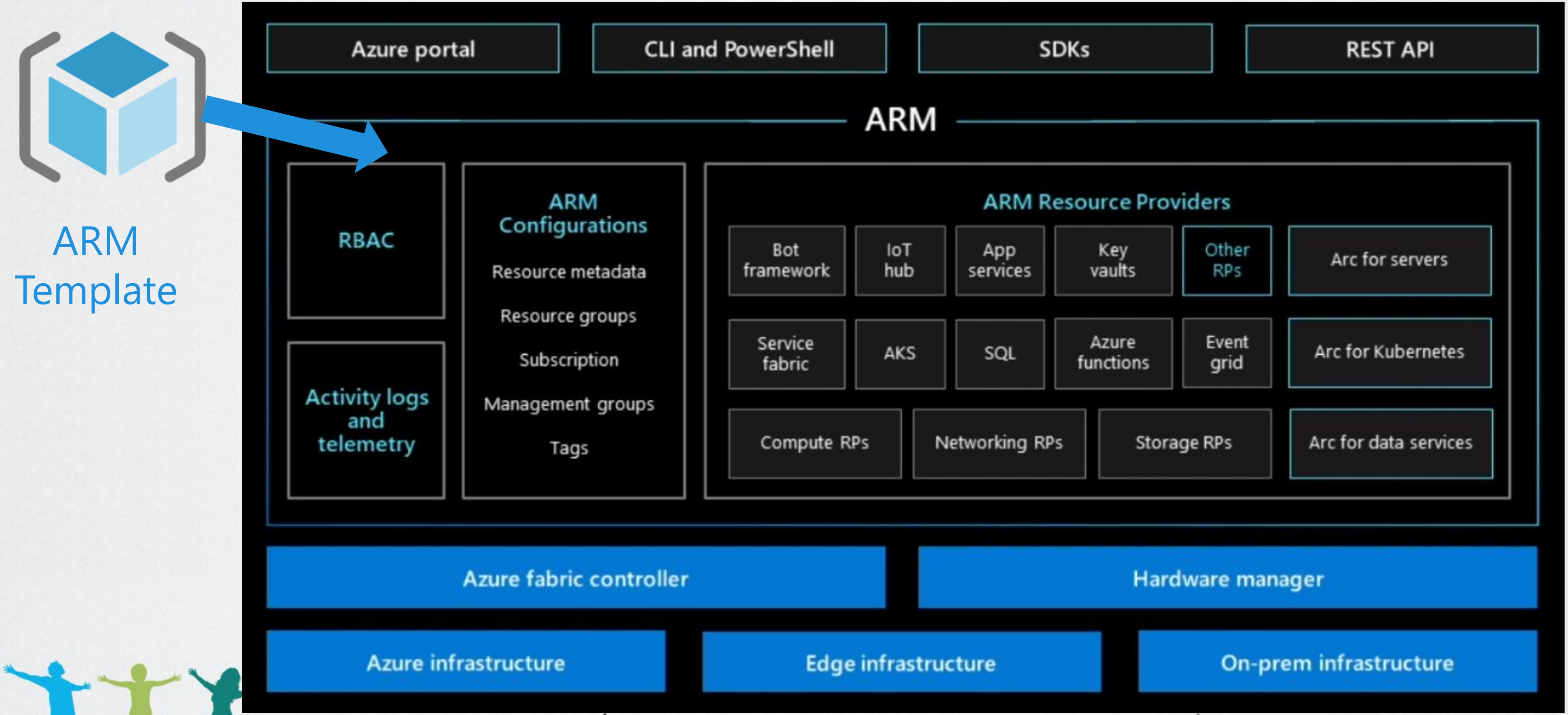
Hardware manager

Azure infrastructure

Edge infrastructure

On-prem infrastructure

Azure Resource Manager



ARM Template

Azure portal | CLI and PowerShell | SDKs | REST API

ARM

RBAC

Activity logs and telemetry

ARM Configurations

- Resource metadata
- Resource groups
- Subscription
- Management groups
- Tags

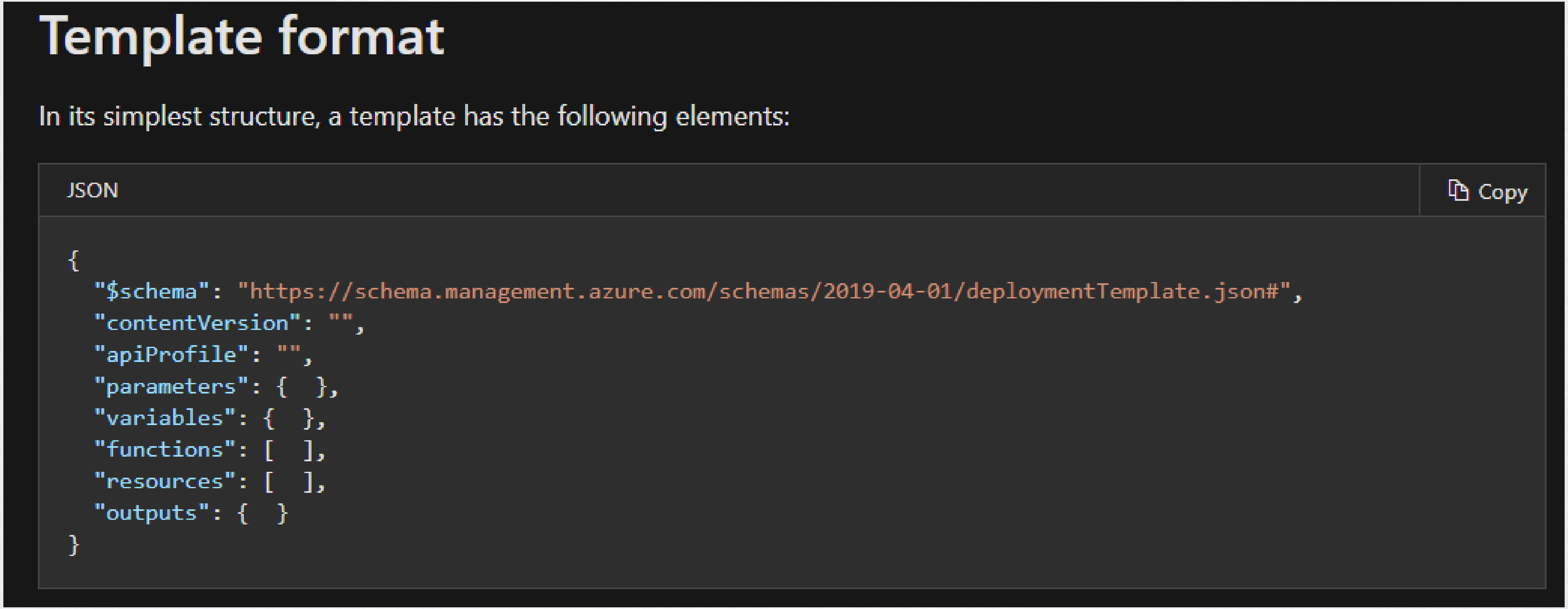
ARM Resource Providers

- Bot framework
- IoT hub
- App services
- Key vaults
- Other RPs
- Arc for servers
- Service fabric
- AKS
- SQL
- Azure functions
- Event grid
- Arc for Kubernetes
- Compute RPs
- Networking RPs
- Storage RPs
- Arc for data services

Azure fabric controller | Hardware manager

Azure infrastructure | Edge infrastructure | On-prem infrastructure

ARM Template

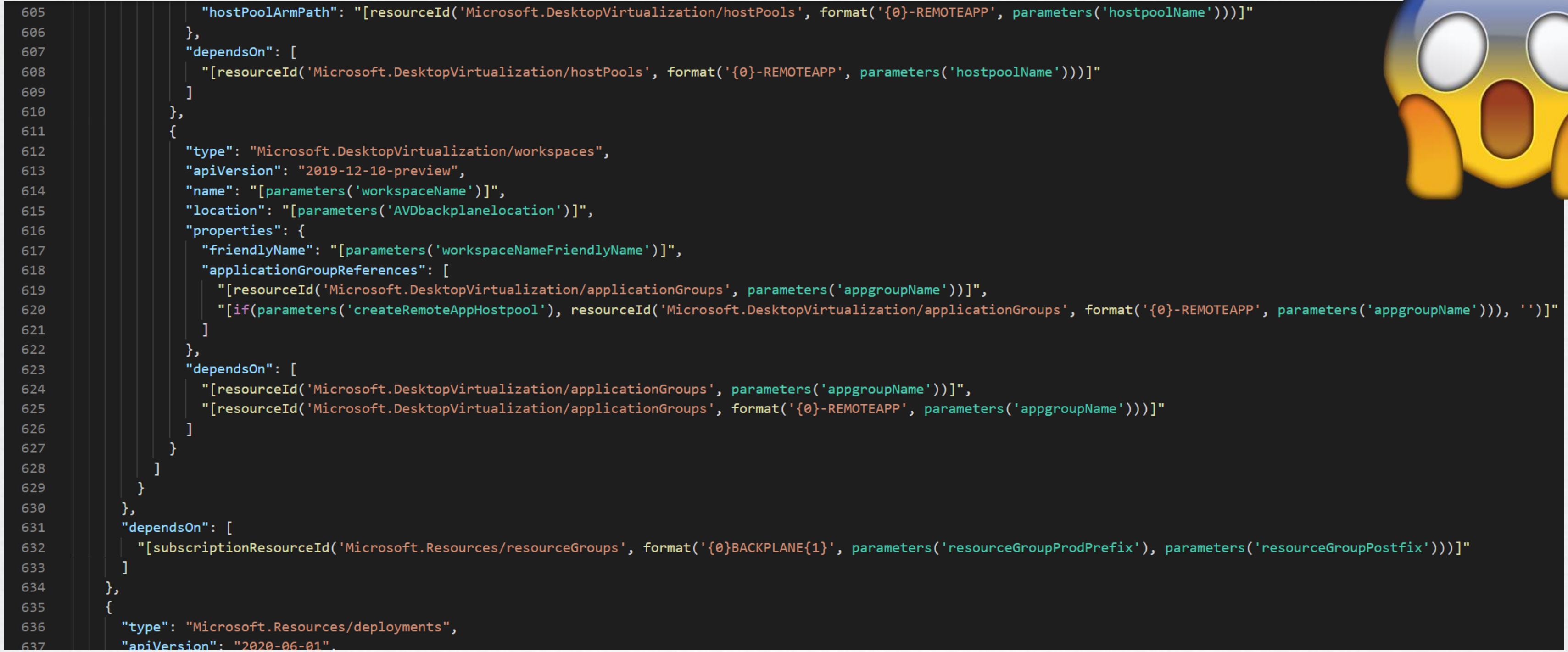


Template format

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

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

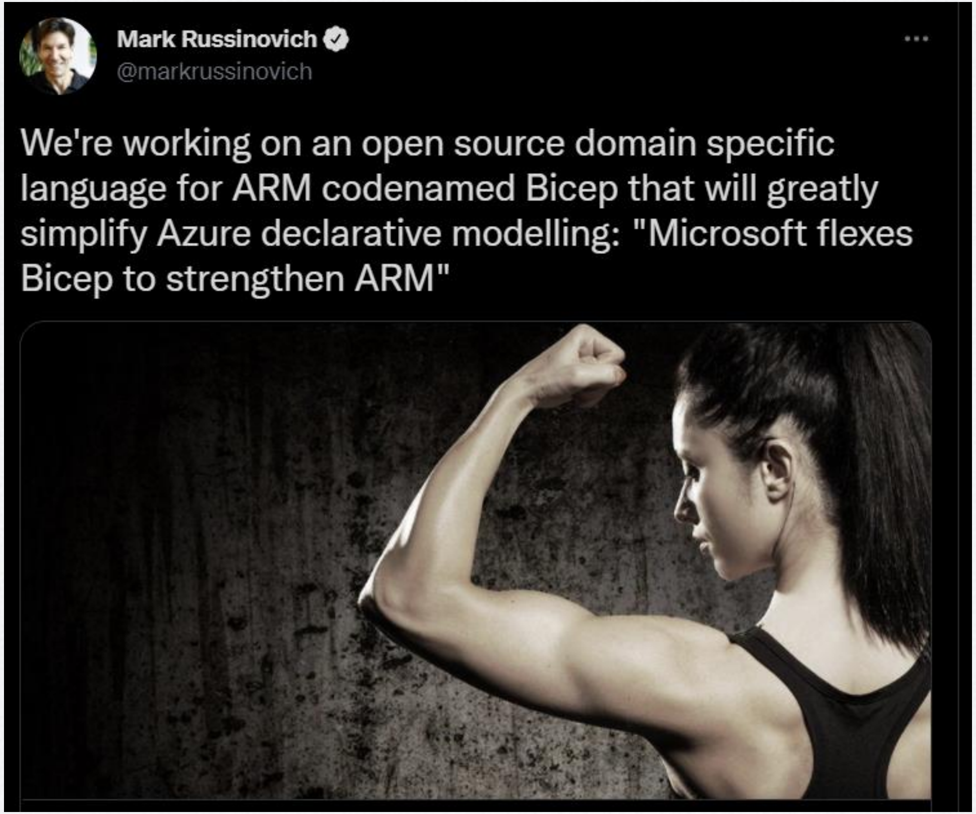
ARM Template complexity



```
                "hostPoolArmPath": "[resourceId('Microsoft.DesktopVirtualization/hostPools', format('{0}-REMOTEAPP', parameters('hostpoolName')))]"
              },
              "dependsOn": [
                "[resourceId('Microsoft.DesktopVirtualization/hostPools', format('{0}-REMOTEAPP', parameters('hostpoolName')))]"
              ]
            },
            {
              "type": "Microsoft.DesktopVirtualization/workspaces",
              "apiVersion": "2019-12-10-preview",
              "name": "[parameters('workspaceName')]",
              "location": "[parameters('AVDbackplanelocation')]",
              "properties": {
                "friendlyName": "[parameters('workspaceNameFriendlyName')]",
                "applicationGroupReferences": [
                  "[resourceId('Microsoft.DesktopVirtualization/applicationGroups', parameters('appgroupName'))]",
                  "[if(parameters('createRemoteAppHostpool'), resourceId('Microsoft.DesktopVirtualization/applicationGroups', format('{0}-REMOTEAPP', parameters('appgroupName'))), '')]"
                ]
              },
              "dependsOn": [
                "[resourceId('Microsoft.DesktopVirtualization/applicationGroups', parameters('appgroupName'))]",
                "[resourceId('Microsoft.DesktopVirtualization/applicationGroups', format('{0}-REMOTEAPP', parameters('appgroupName')))]"
              ]
            }
          ]
        }
      },
      "dependsOn": [
        "[subscriptionResourceId('Microsoft.Resources/resourceGroups', format('{0}BACKPLANE{1}', parameters('resourceGroupProdPrefix'), parameters('resourceGroupPostfix')))]"
      ]
    },
    {
      "type": "Microsoft.Resources/deployments",
      "apiVersion": "2020-06-01",
```

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

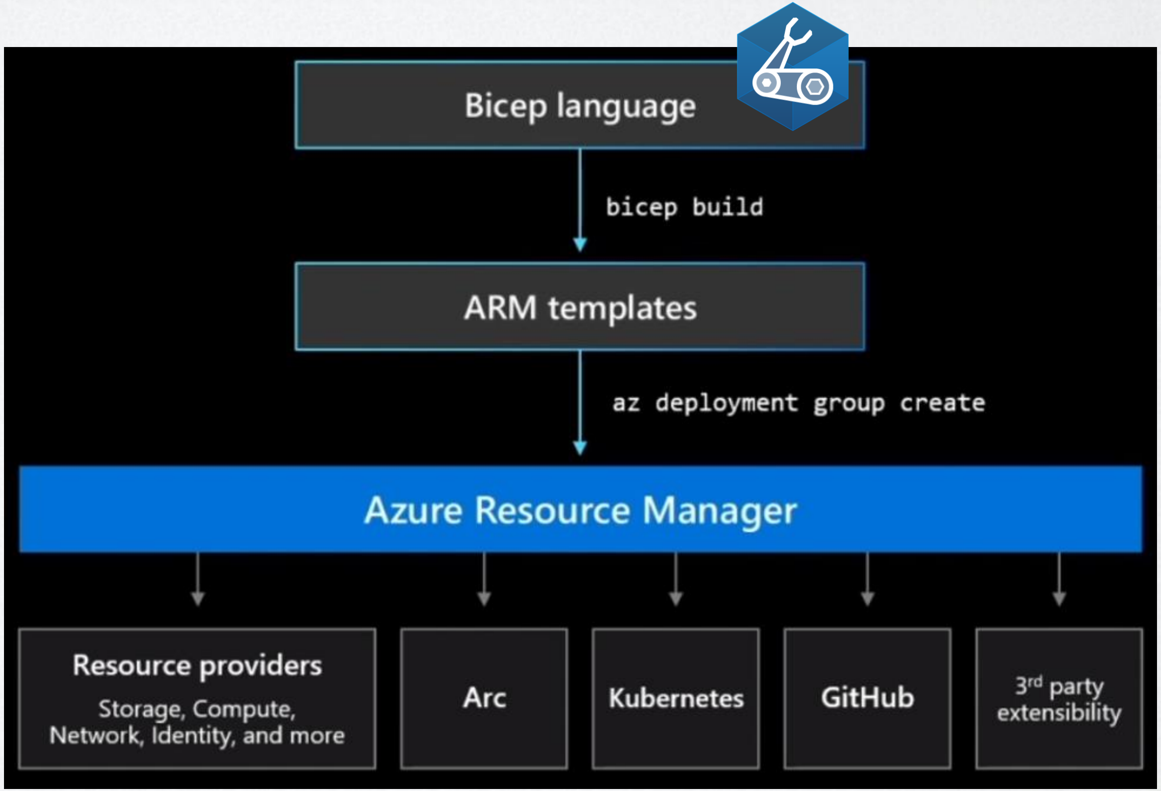
Leverage ARM template knowledge and investments

Modular

Abstract common blocks of code into reusable parts

Open Source

Transparency and community

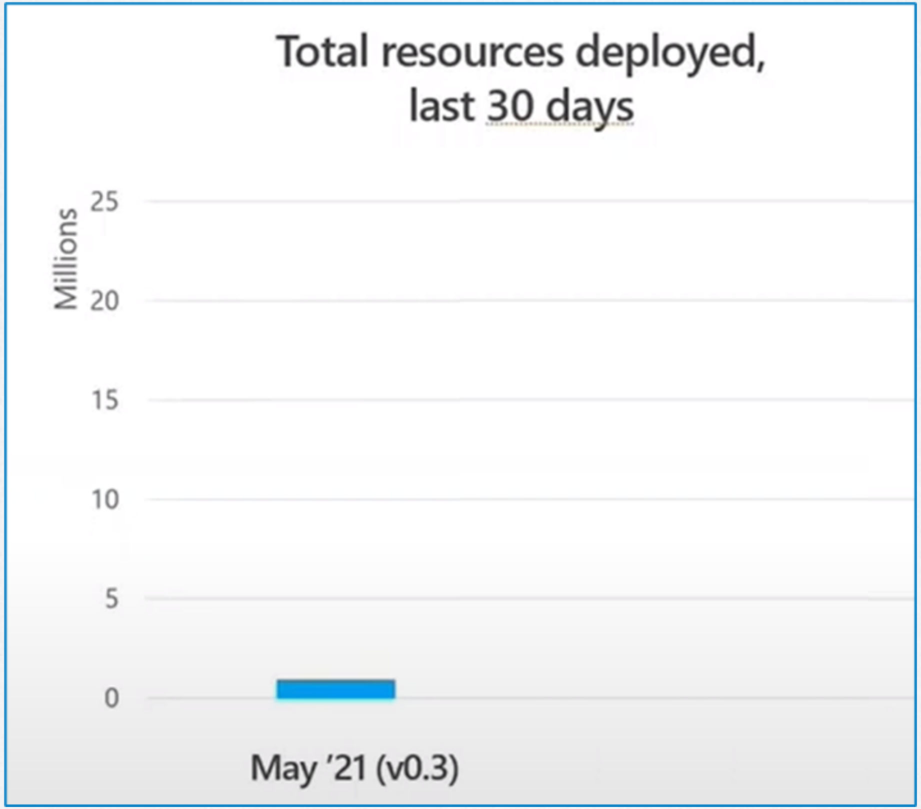


```
bicep build
```

```
az deployment group create
```

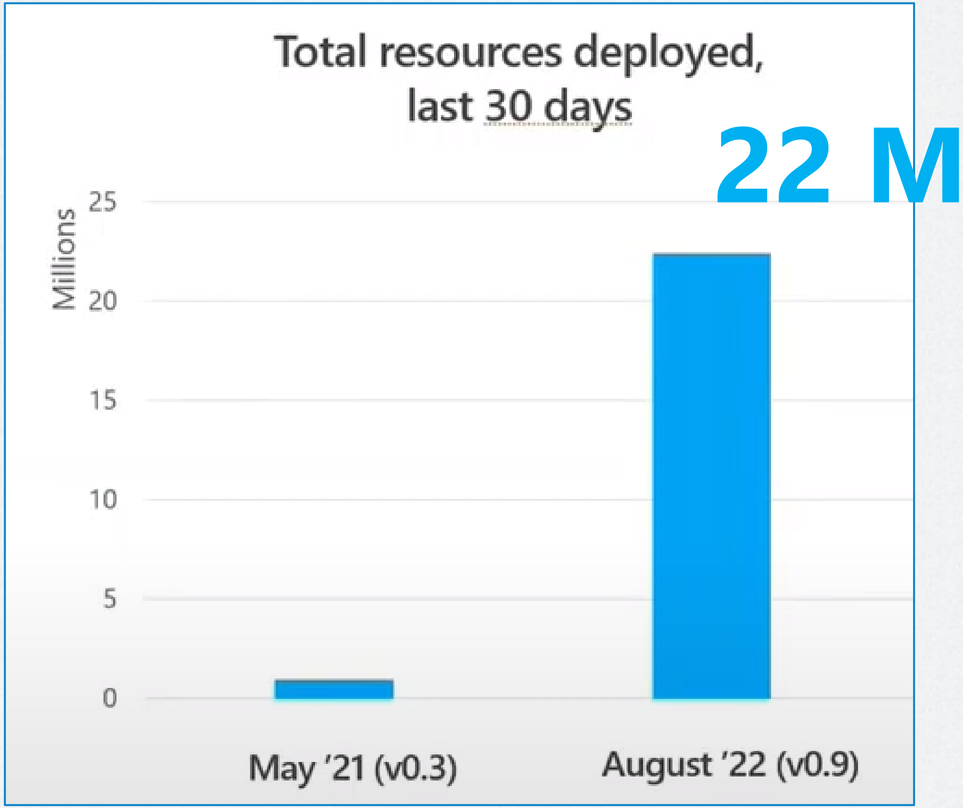
DEVOPS

What is the Bicep usage?



DEVOPS

What is the Bicep usage?



22 Million!!!

Delta-N Connecting the Cloud cegeka ARROW LIQUIT INSPARK Microsoft

Getting started

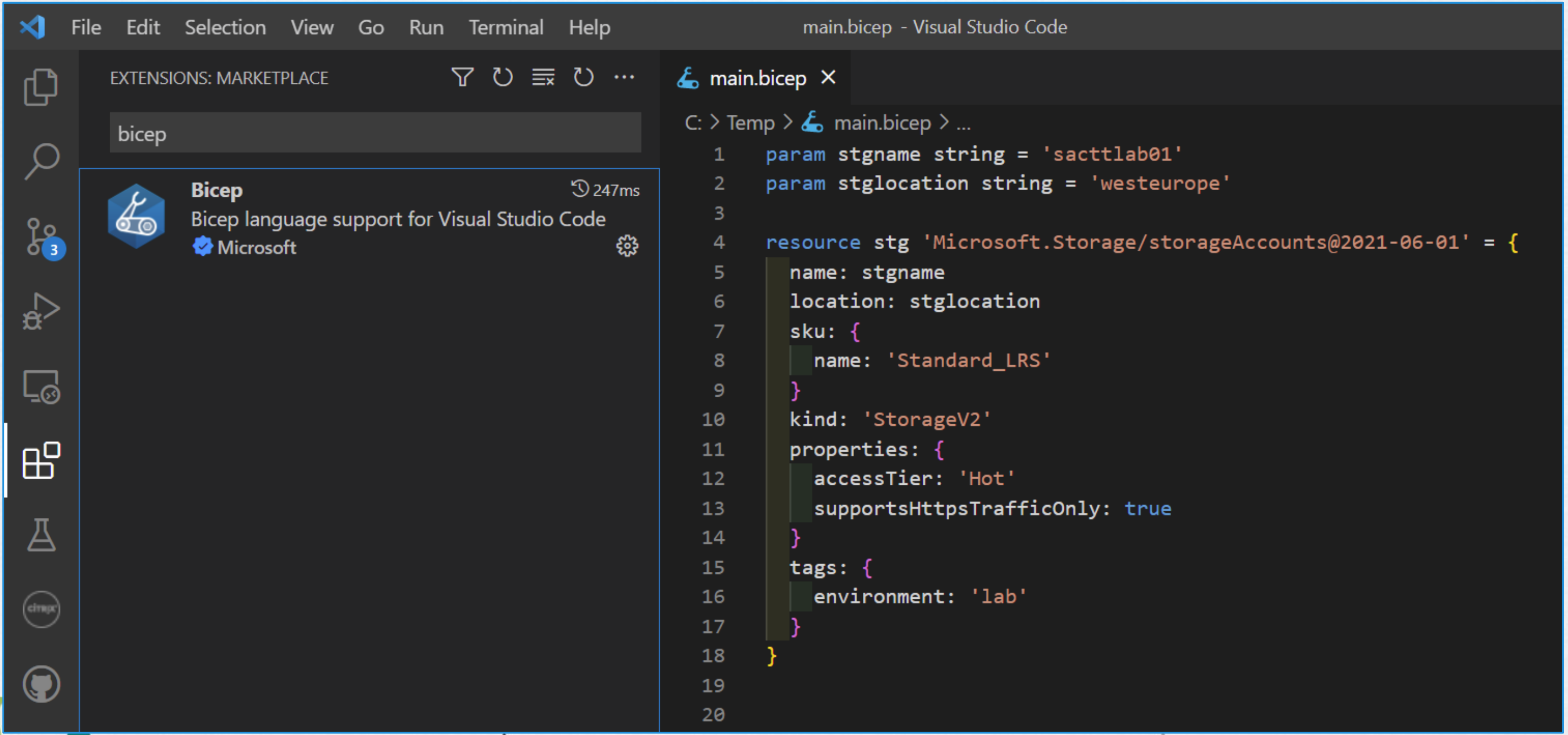
1. Install the tools (bicep.exe, Visual Studio Code, PowerShell/az cli)

+
+
/
Visual Studio Code

DEVOPS

Getting started

2. Create a Bicep template

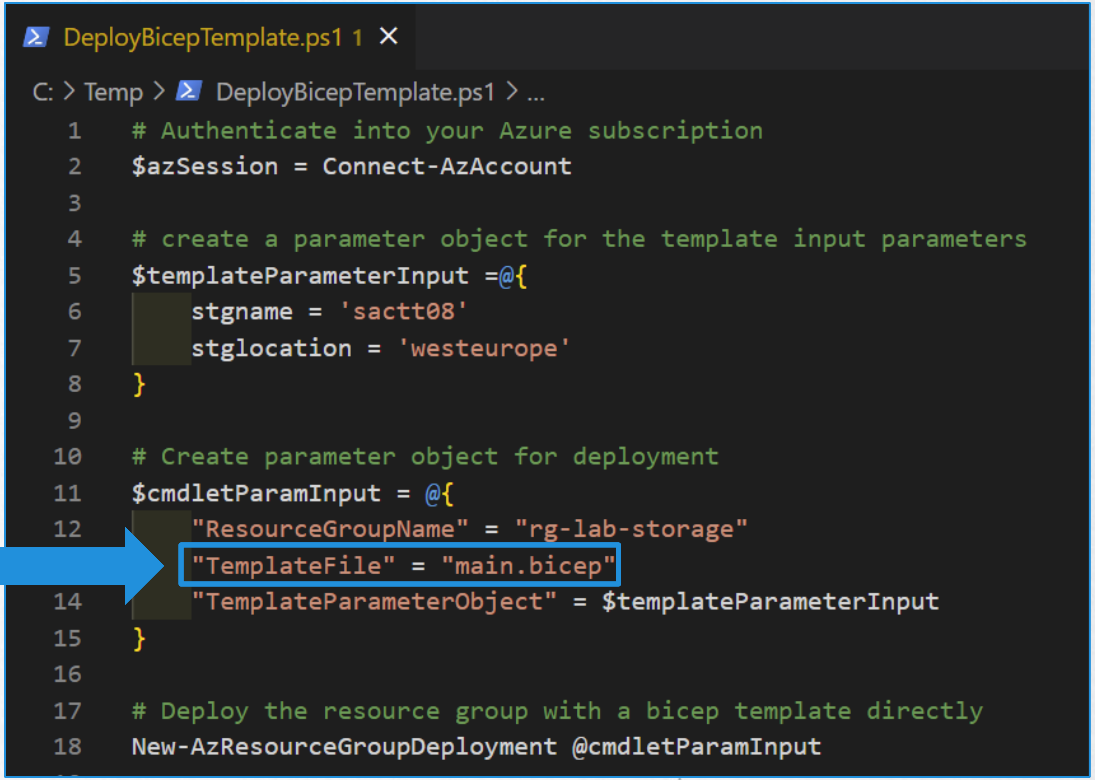

File
Edit
Selection
View
Go
Run
Terminal
Help
main.bicep - Visual Studio Code
EXTENSIONS: MARKETPLACE
bicep
Bicep
247ms
Bicep language support for Visual Studio Code
Microsoft
3
main.bicep
C: > Temp > main.bicep > ...

```
param stgname string = 'sacttlab01'
param stglocation string = 'westeurope'

resource stg 'Microsoft.Storage/storageAccounts@2021-06-01' = {
  name: stgname
  location: stglocation
  sku: {
    name: 'Standard_LRS'
  }
  kind: 'StorageV2'
  properties: {
    accessTier: 'Hot'
    supportsHttpsTrafficOnly: true
  }
  tags: {
    environment: 'lab'
  }
}


```

DEVOPS

Getting started

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

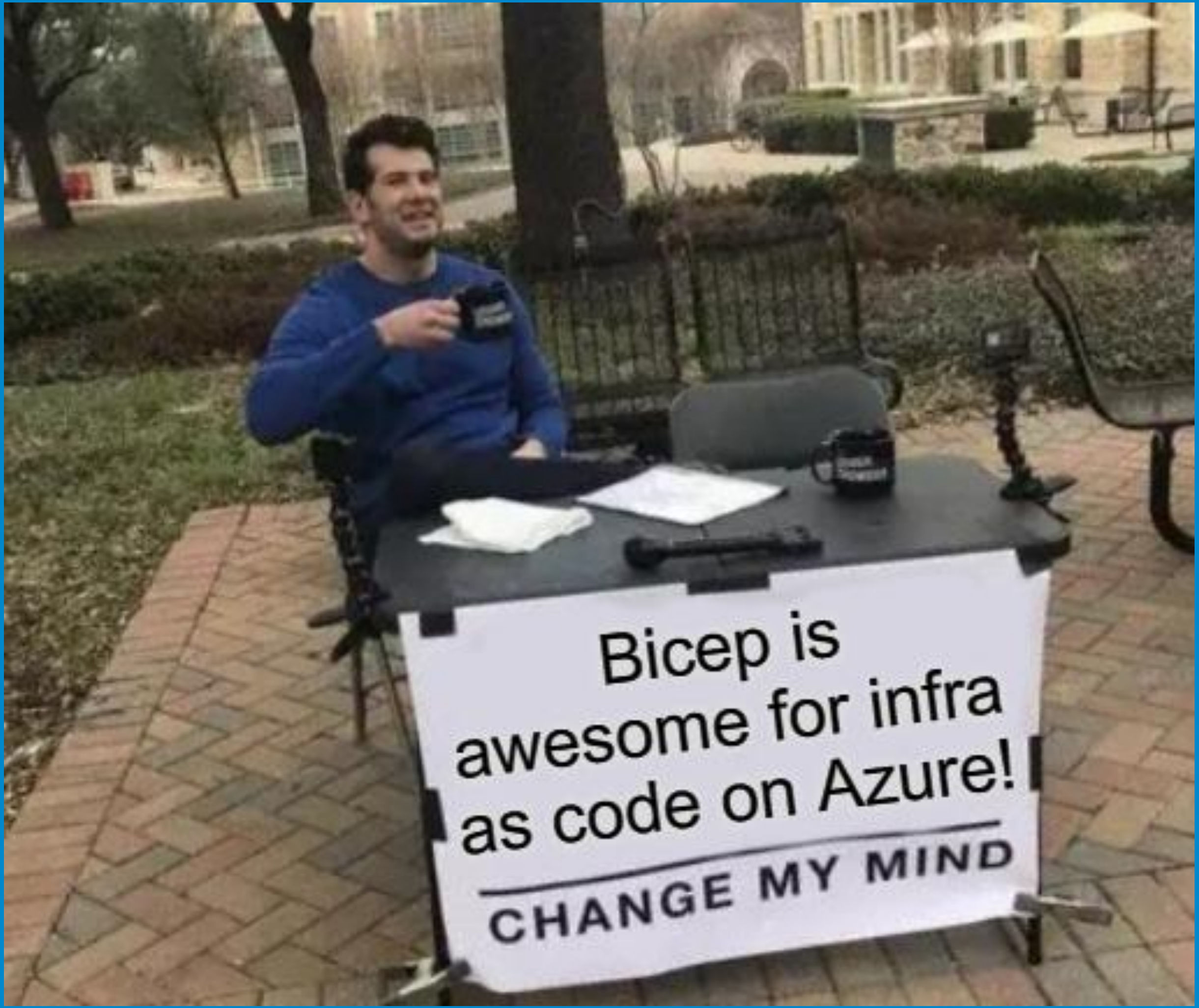

DeployBicepTemplate.ps1 1 ×
C: > Temp > DeployBicepTemplate.ps1 > ...

```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
```

DEVOPS

DEMO!

Getting started with Bicep code (and Vscode)



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

Next up!

~~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