Empowering ARM and JSON with 'Project Bicep'

Esther Barthel

@virtuEs_IT

github.com/cognitionit

Microsoft MVP

Freek Berson

@fberson

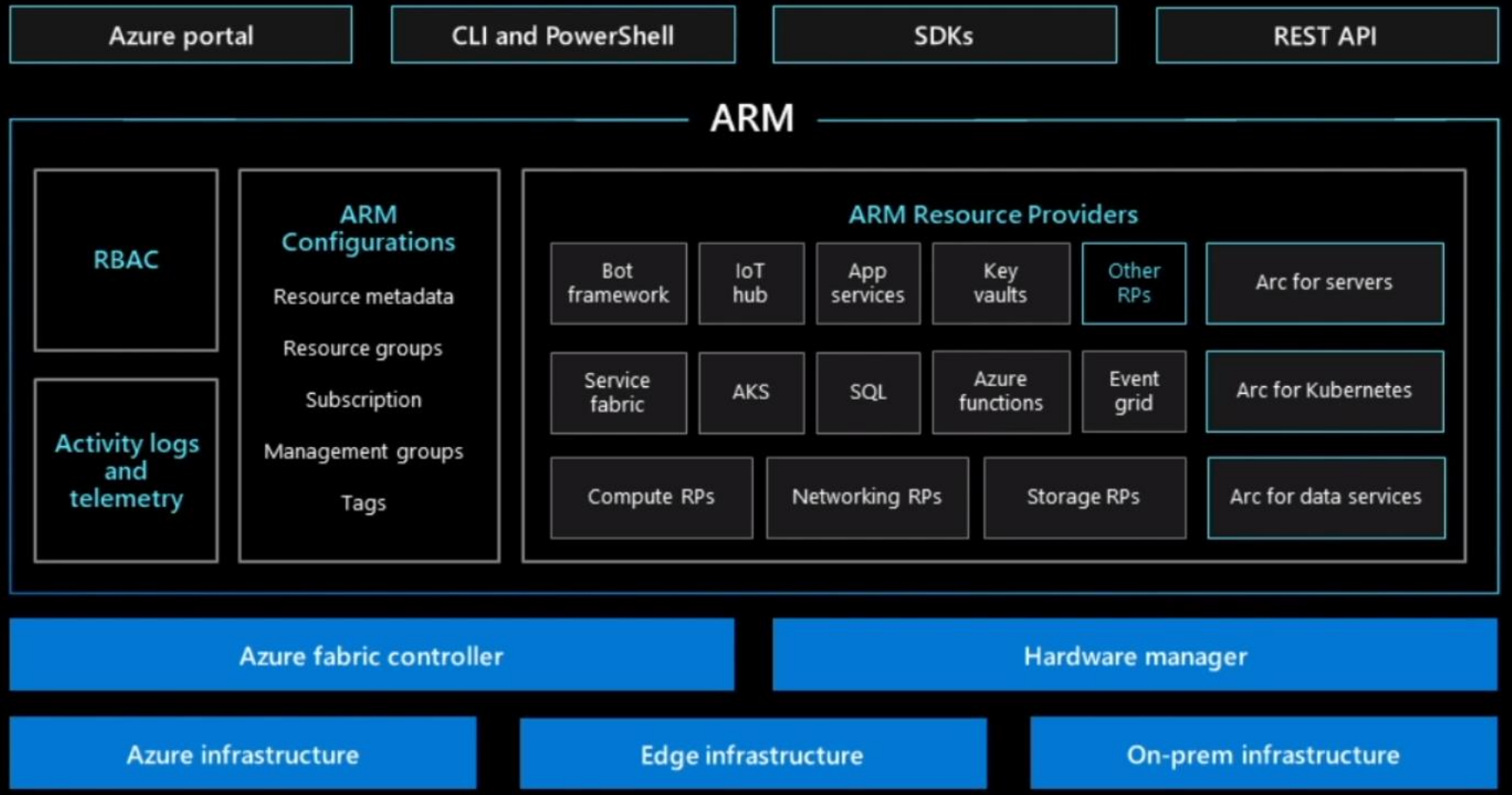
github.com/fberson

Microsoft MVP

Agenda

- Introduction to automation
- Azure Resource Manager & JSON
- What is Project ‘Bicep’
- Demos, demos, demos!

Azure Resource Manager



How to get started with ARM & JSON

… structure of an **Azure Resource Manager template**.

The template consists of **JSON** and expressions that you can use to construct values for your deployment.

ARM Template

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

```
"parameters": {
  "<parameter-name>" : {
    "type" : "<type-of-parameter-value>",
    "defaultValue": "<default-value-of-parameter>",
    "allowedValues": [ "<array-of-allowed-values>" ],
    "minValue": <minimum-value-for-int>,
    "maxValue": <maximum-value-for-int>,
    "minLength": <minimum-length-for-string-or-array>,
    "maxLength": <maximum-length-for-string-or-array-parameters>,
    "metadata": {
      "description"
    }
  }
}
```

```
"variables": {
  "<variable-name>": "<variable-value>",
  "<variable-name>": {
    <variable-complex-type-value>
  },
  "<variable-object-name>": {
    "copy": [
      {
        "name": "<name-of-array-property>",
        "count": <number-of-iterations>,
        "input": <object-or-value-to-repeat>
```

```
[

me": "<variable-array-name>",
unt": <number-of-iterations>,
put": <object-or-value-to-repeat>
```

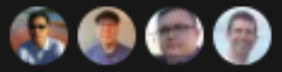
```
"functions": [
  {
    "namespace": "<namespace-for-functions>",
    "members": {
      "<function-name>": {
        "parameters": [
          {
            "name": "<parameter-name>",
            "type": "<type-of-parameter-value>"
          }
        ],
        "output": {
          "type": "<type-of-output-value>",
          "value": "<function-return-value>"
        }
      }
    }
  }
],
```

```
"outputs": {
  "<output-name>": {
    "condition": "<boolean-value-whether-to-output-value>",
    "type": "<type-of-output-value>",
    "value": "<output-value-expression>",
    "copy": {
      "count": <number-of-iterations>,
      "input": <values-for-the-variable>
    }
  }
}
```

```
"resources": [
  {
    "condition": "<true-to-deploy-this-resource>",
    "type": "<resource-provider-namespace/resource-type-name>",
    "apiVersion": "<api-version-of-resource>",
    "name": "<name-of-the-resource>",
    "comments": "<your-reference-notes>",
    "location": "<location-of-resource>",
    "dependsOn": [
      "<array-of-related-resource-names>"
    ],
    "tags": {
      "<tag-name1>": "<tag-value1>",
      "<tag-name2>": "<tag-value2>"
    },
    "sku": {
      "name": "<sku-name>",
      "tier": "<sku-tier>",
      "size": "<sku-size>",
      "family": "<sku-family>",
      "capacity": <sku-capacity>
    },
    "kind": "<type-of-resource>",
    "copy": {
      "name": "<name-of-copy-loop>",
      "count": <number-of-iterations>,
      "mode": "<serial-or-parallel>",
      "batchSize": <number-to-deploy-serially>
    },
    "plan": {
      "name": "<plan-name>",
      "promotionCode": "<plan-promotion-code>",
      "publisher": "<plan-publisher>",
      "product": "<plan-product>",
```

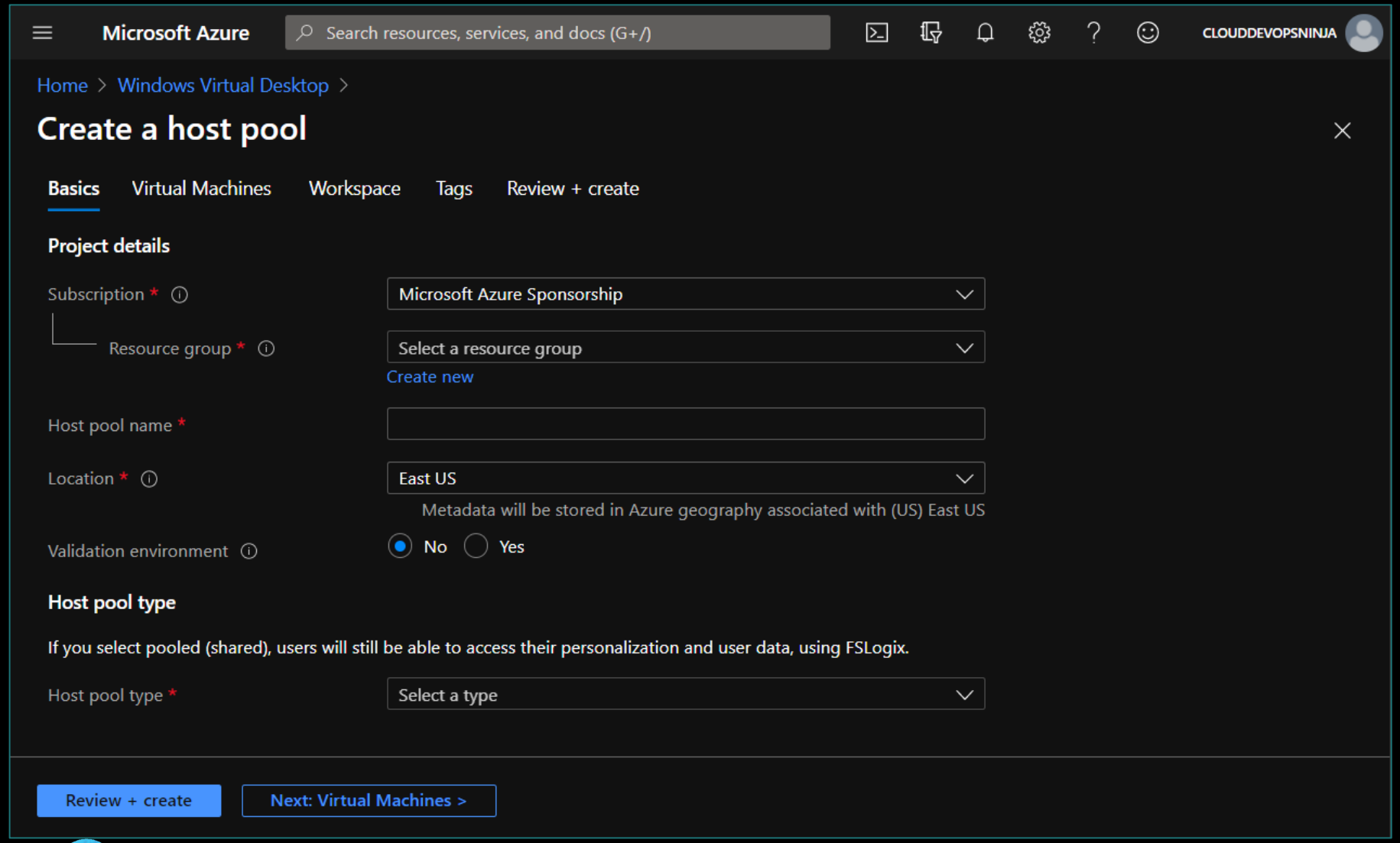
ARM Template

Quickstart: Create and deploy ARM templates by using the Azure portal

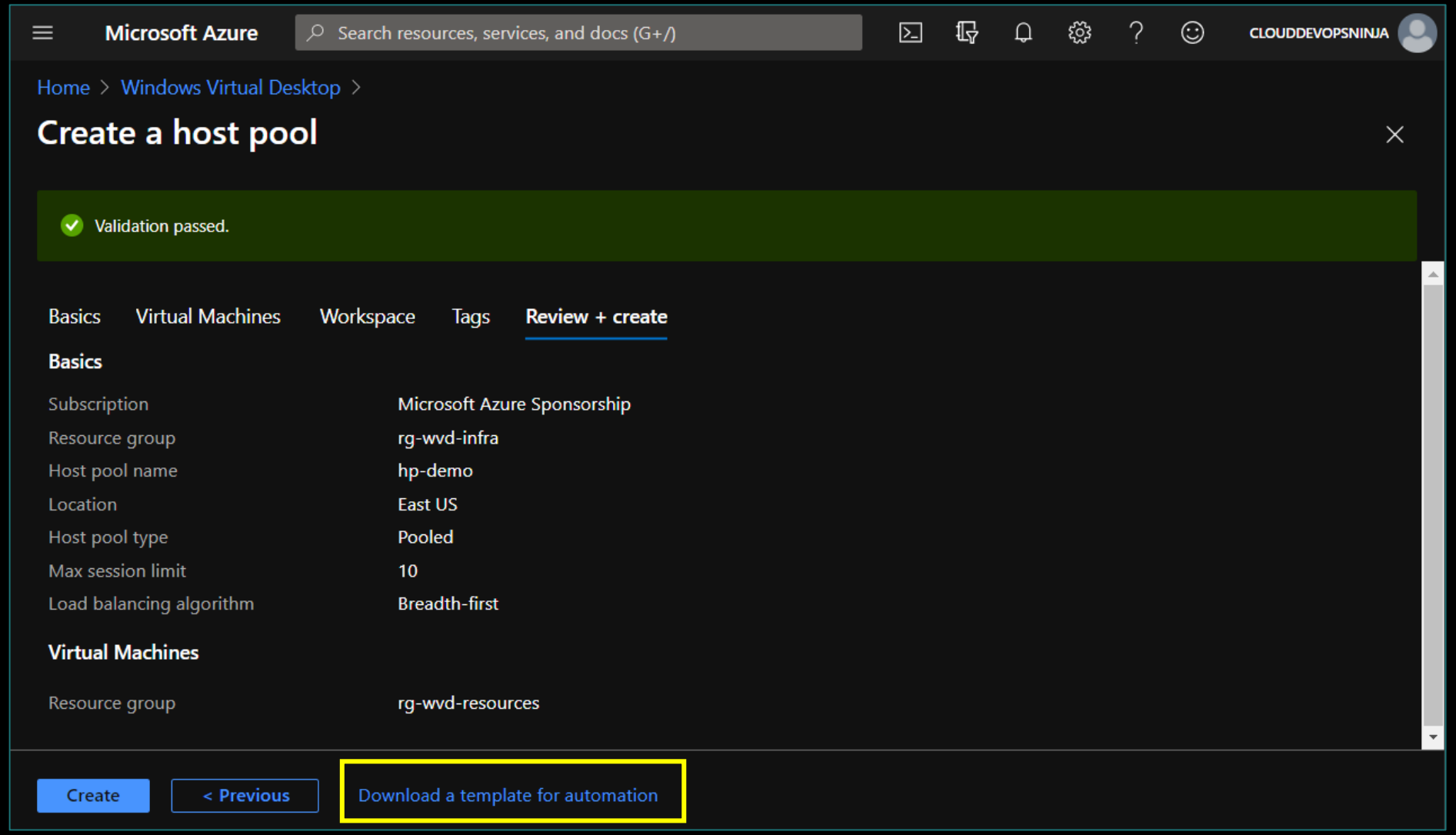
06/29/2020 • 6 minutes to read • 

Learn how to generate an Azure Resource Manager template (ARM template) using the Azure portal, and the process of editing and deploying the template from the portal. ARM templates are JSON files that define the resources you need to deploy for your solution. To understand the concepts associated with deploying and managing your Azure solutions, see template deployment overview.

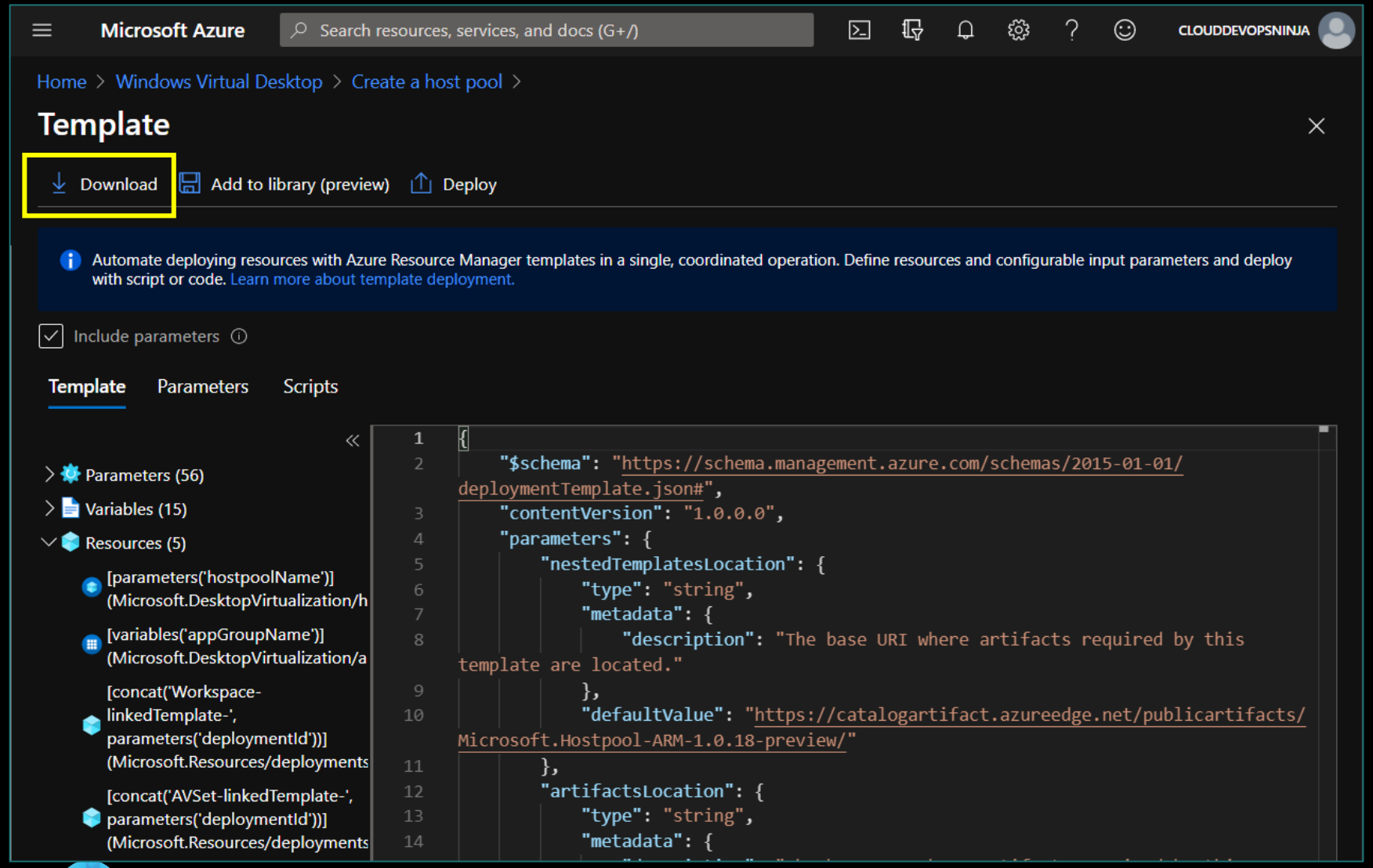
ARM Template



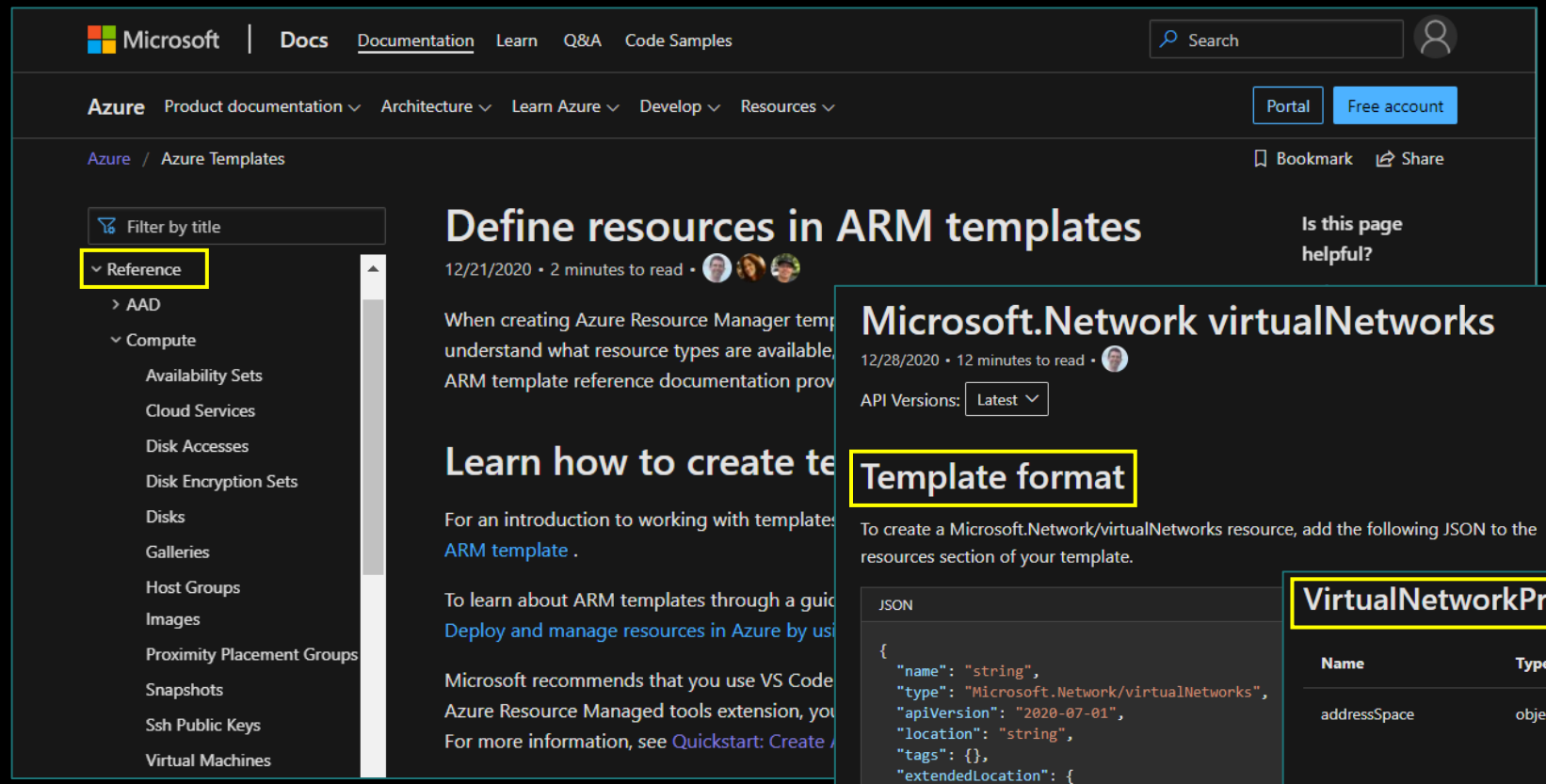
ARM Template

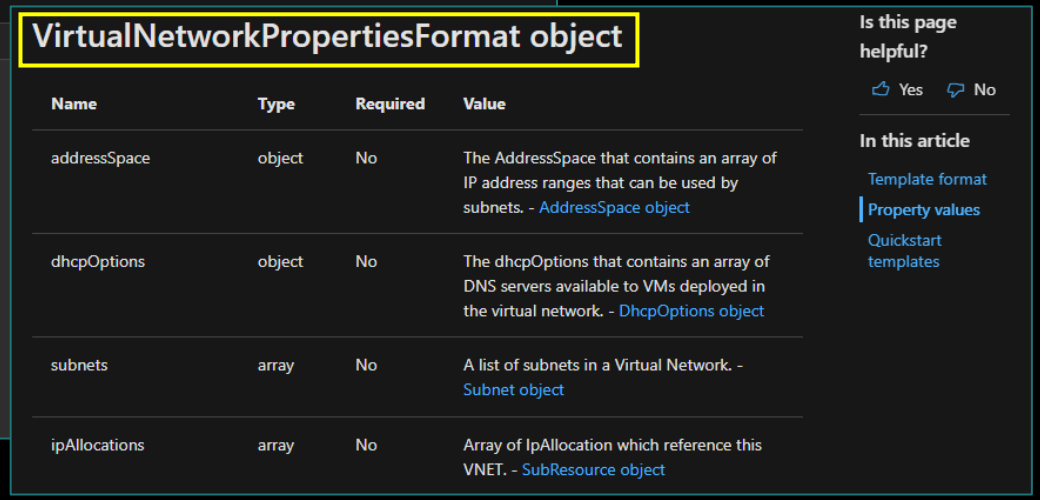


ARM Template



ARM Template – Reference Guide





ARM Template – Reference Guide

```
"parameters": {
    "dnsServer": {
        "type": "string",
        "metadata": {
            "description": "The vNet DNS server to enable the domain-join of Session Hosts"
        }
    }
},
"resources": [
    {
        1 child: string (subnets)
        "name": "[variables('vnetName')]",
        "type": "Microsoft.Network/virtualNetworks",
        "apiVersion": "2020-07-01",
        "location": "[resourceGroup().location]",
        "properties": {
            "addressSpace": {"addressPrefixes": ["10.0.0.0/16"]},
            "dhcpOptions": {"dnsServers": ["parameters('dnsServer')"]},
```

ARM Template learning path

https://bit.ly/3qZGNj1

MODULE 1: Deploy Azure infrastructure by using ARM templates

MODULE 2: Deploy to multiple Azure environments by using ARM template features

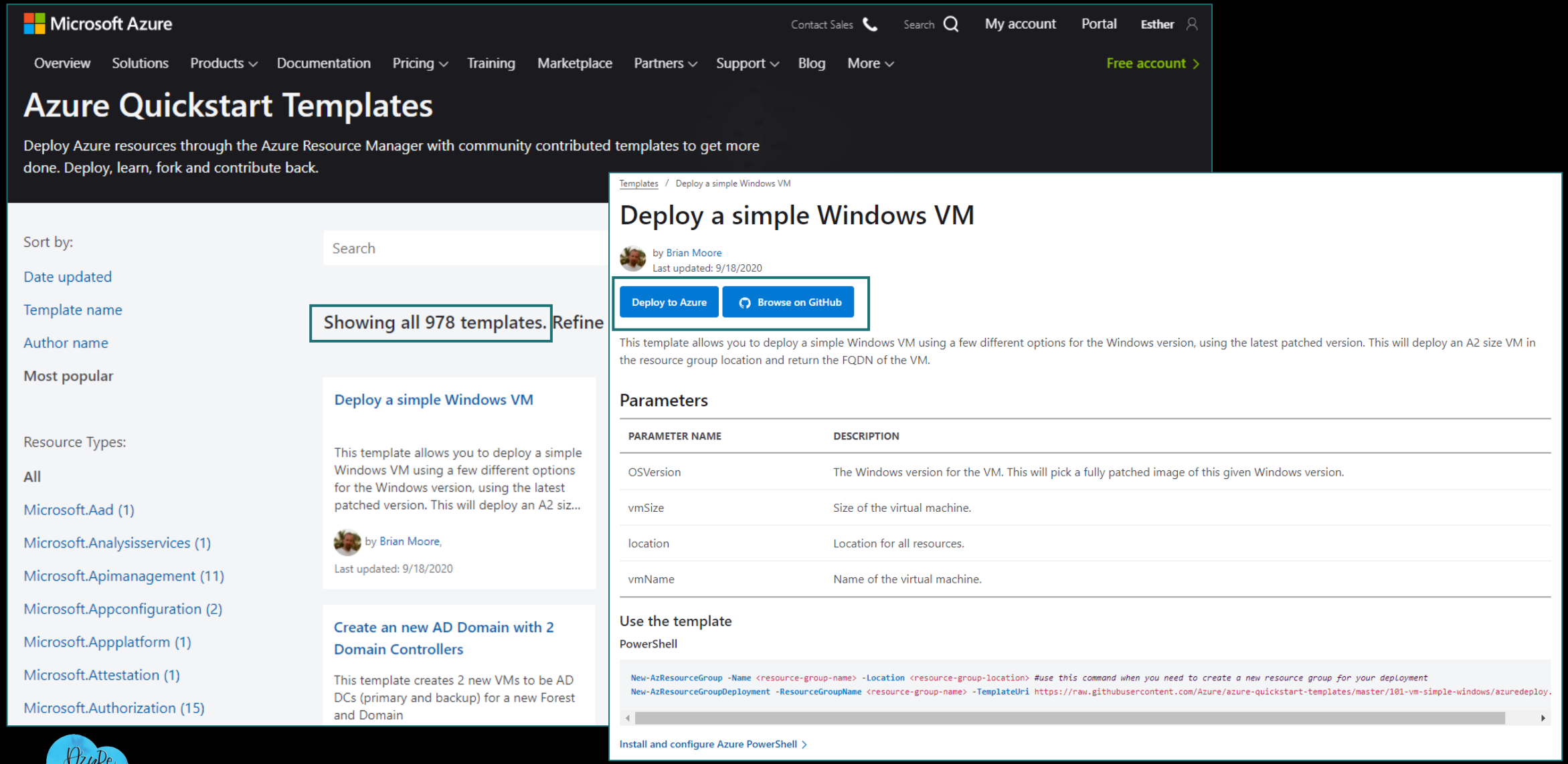
MODULE 3: Preview changes and validate Azure resources by using what-if and the ARM template test toolkit

MODULE 4: Automate the deployment of ARM templates by using GitHub Action

MODULE 5: Extend ARM templates by using deployment scripts

MODULE 6: Manage complex cloud deployments by using advanced ARM template features

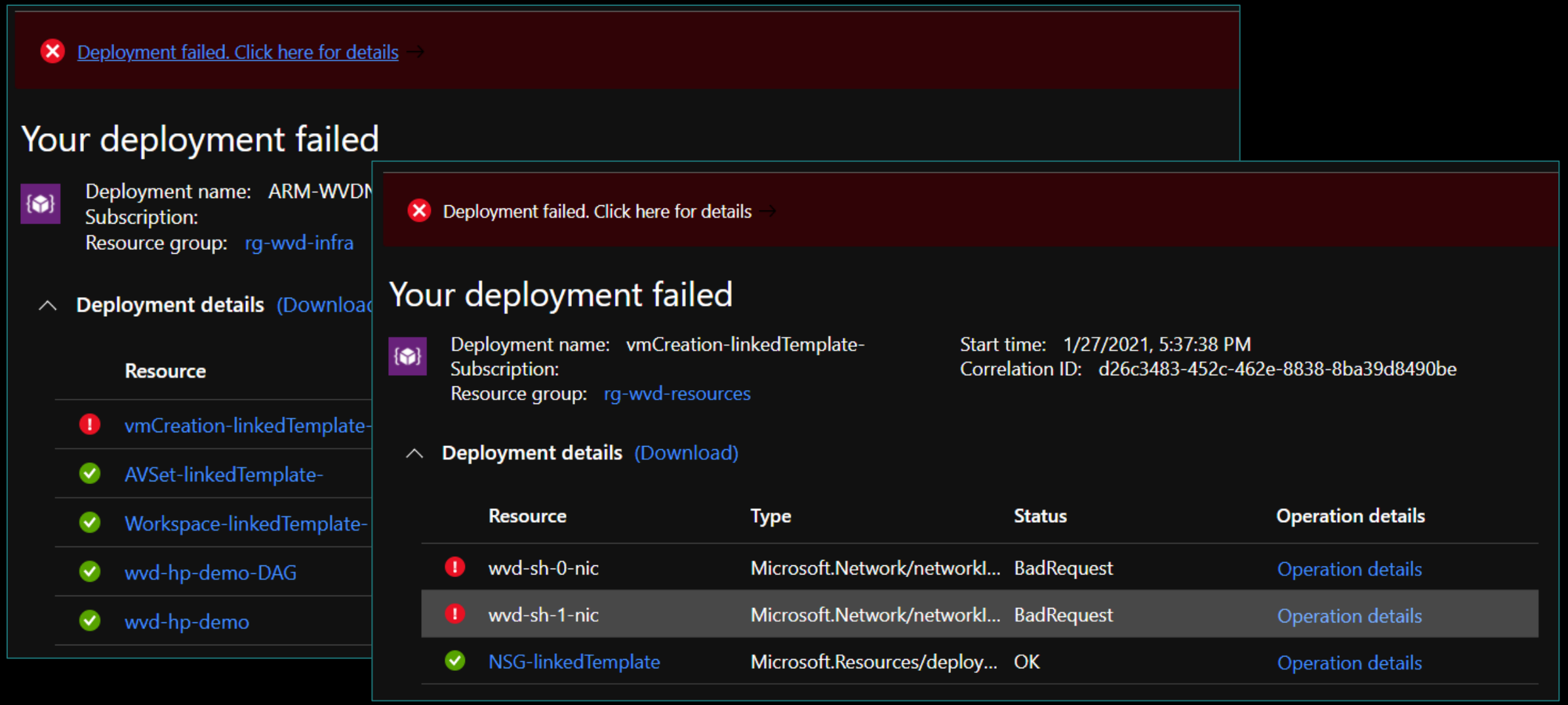
ARM Template – Azure Quickstart Templates



ARM Template – Automatic deployments

```powershell
# ARM Template file
$jsonARMTemplateFile = ".\ARM-WVDNewHostpool.template.json"
$jsonARMParameterFile = ".\ARM-WVDNewHostpool.parameter.json"

# Create WVD Hostpool, based on ARM Template
New-AzResourceGroupDeployment -ResourceGroupName "rg-wvd-infra" `
    -TemplateFile $jsonARMTemplateFile `
    -TemplateParameterFile $jsonARMParameterFile `
    -administratorAccountPassword $secureDomainAdminPassword `
    -vmAdministratorAccountPassword $secureLocalAdminPassword `
    -Verbose
```

ARM Template – Automatic deployments



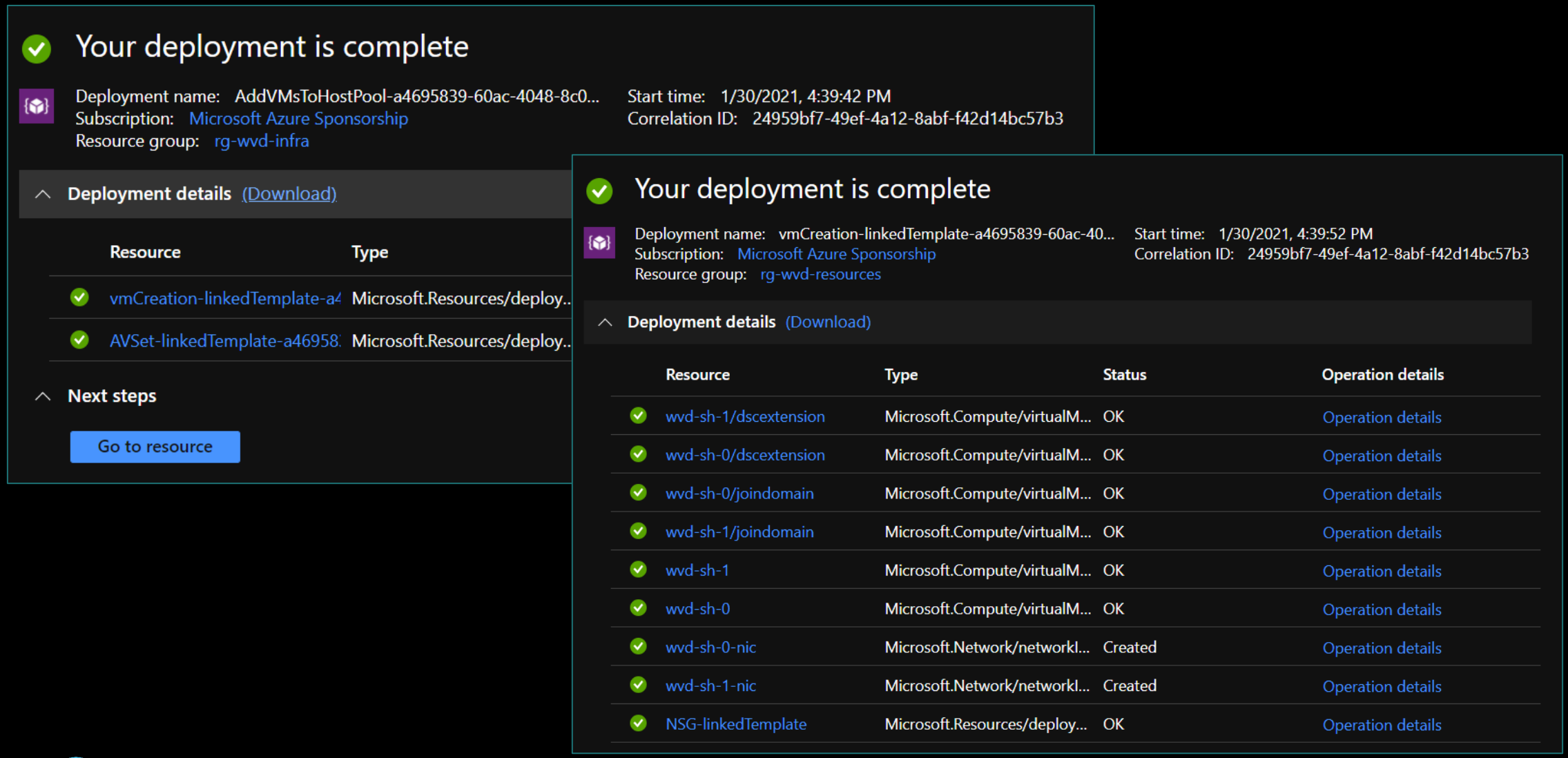
ARM Template – Automatic deployments


```
{
  "code": "DeploymentFailed",
  "message": "At least one resource deployment operation failed. Please list deployment operations for details. Please see https://aka.ms/DeployOperations for usage details.",
  "details": [
    {
      "code": "InvalidResourceReference",
      "message": "Resource /subscriptions/
                                  /resourceGroups/
rg-wvd-resources/providers/Microsoft.Network/
virtualNetworks/vnet-wvd-resource/subnets/default
referenced by resource /subscriptions/
                                  /resourceGroups/
rg-wvd-resources/providers/Microsoft.Network/
networkInterfaces/wvd-sh-0-nic was not found. Please make
sure that the referenced resource exists, and that both
resources are in the same region."
    },
```

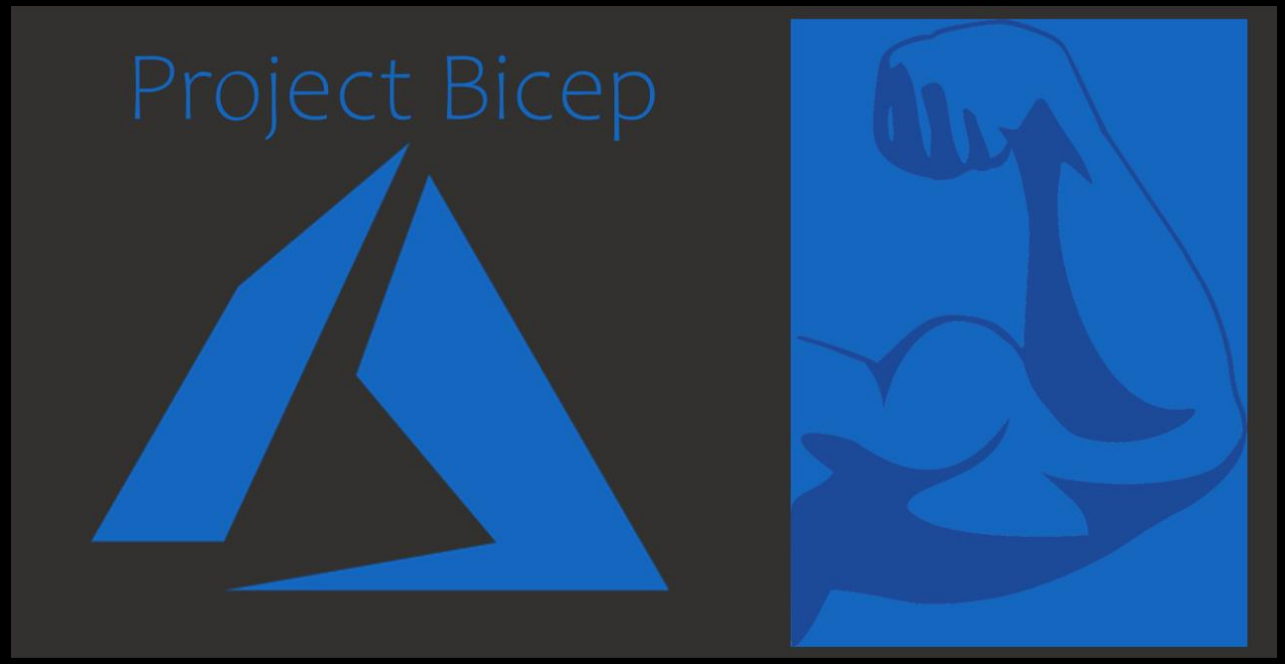
ARM Template – Automatic deployments

- 531 lines of code
- complex JSON formatting
- advanced options:
  - nested templates
  - linked templates

ARM Template – Automatic deployments



What is Project ‘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.

Project 'Bicep'

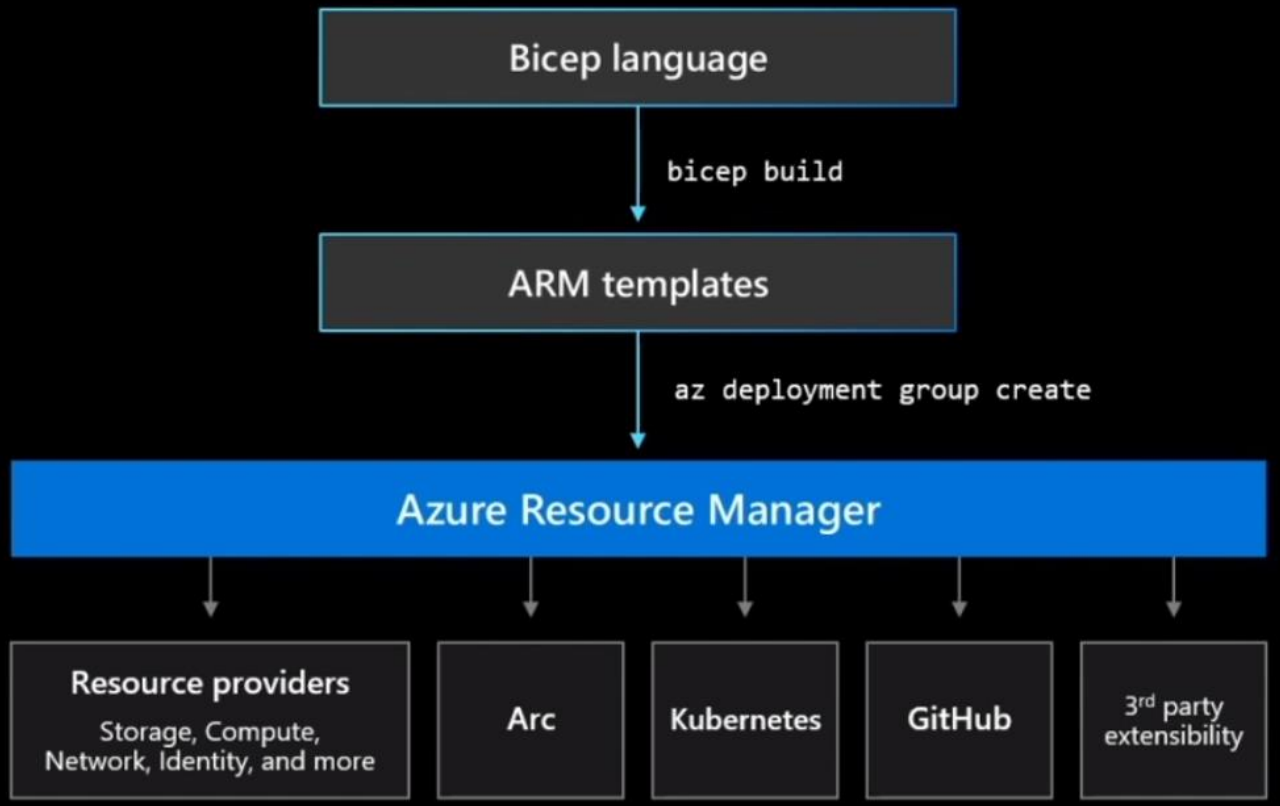
Simple declarative language to provision infrastructure to Azure.

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



Installing project 'Bicep'

1. Install the Bicep CLI (required)

```
# Create the install folder
$installPath = "$env:USERPROFILE\.bicep"
$installDir = New-Item -ItemType Directory -Path $installPath -Force
$installDir.Attributes += 'Hidden'
# Fetch the latest Bicep CLI binary
(New-Object Net.WebClient).DownloadFile("https://github.com/Azure/bicep/releases/latest/download/bicep-win-x64.exe", "$installPath\bicep.exe")
# Add bicep to your PATH
$currentPath = (Get-Item -path "HKCU:\Environment" ).GetValue('Path', '', 'DoNotExpandEnvironmentNames')
if (-not $currentPath.Contains("%USERPROFILE%\.bicep")) { setx PATH ($currentPath + ";%USERPROFILE%\.bicep") }
if (-not $env:path.Contains($installPath)) { $env:path += ";$installPath" }
# Verify you can now access the 'bicep' command.
bicep --help
# Done!
```

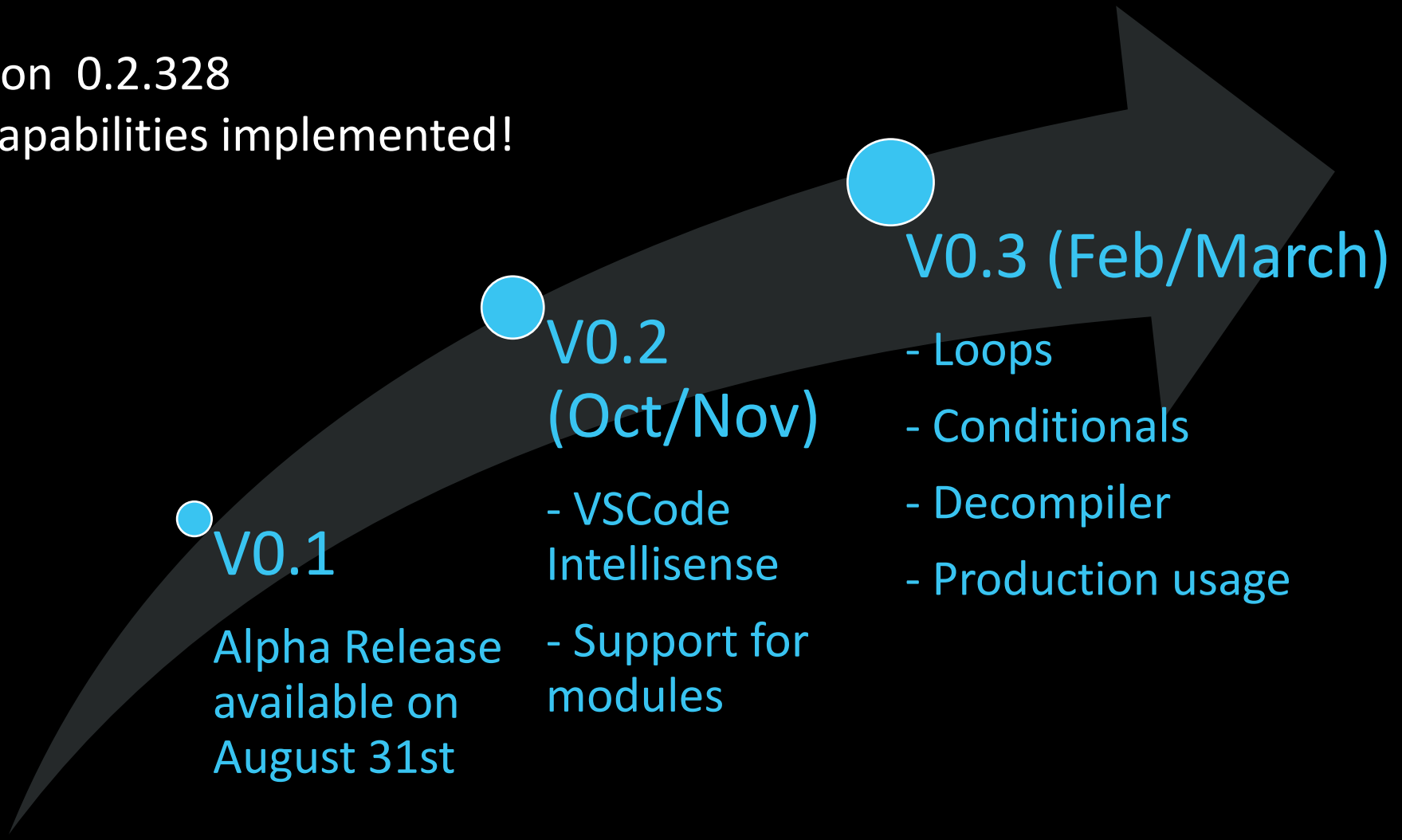
2. Install the Bicep VS Code extension(optional)

```
# Fetch the latest Bicep VSCode extension
$vsixPath = "$env:TEMP\vscode-bicep.vsix"
(New-Object Net.WebClient).DownloadFile("https://github.com/Azure/bicep/releases/latest/download/vscode-bicep.vsix", $vsixPath)
# Install the extension
code --install-extension $vsixPath
# Clean up the file
Remove-Item $vsixPath
# Done!
```

Demo

Road map

Current release: CLI version 0.2.328
~90% of ARM template capabilities implemented!



Breaking changes planned

BEFORE

AFTER

Param Decorators

Issue #64

```
param authType string {
  default: 'password'
  allowed: [
    'sshPublicKey'
    'password'
  ]
}
```

```
@allowed([
  'sshPublicKey'
  'password'
])
param authType string = 'password'
```

Outputs

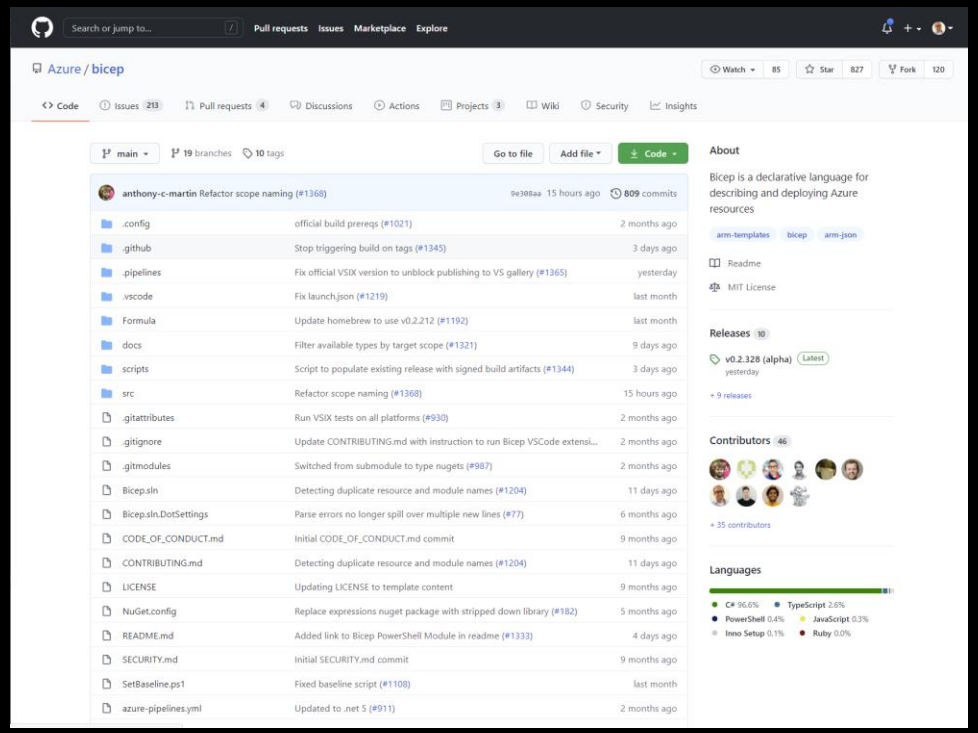
Issue #569

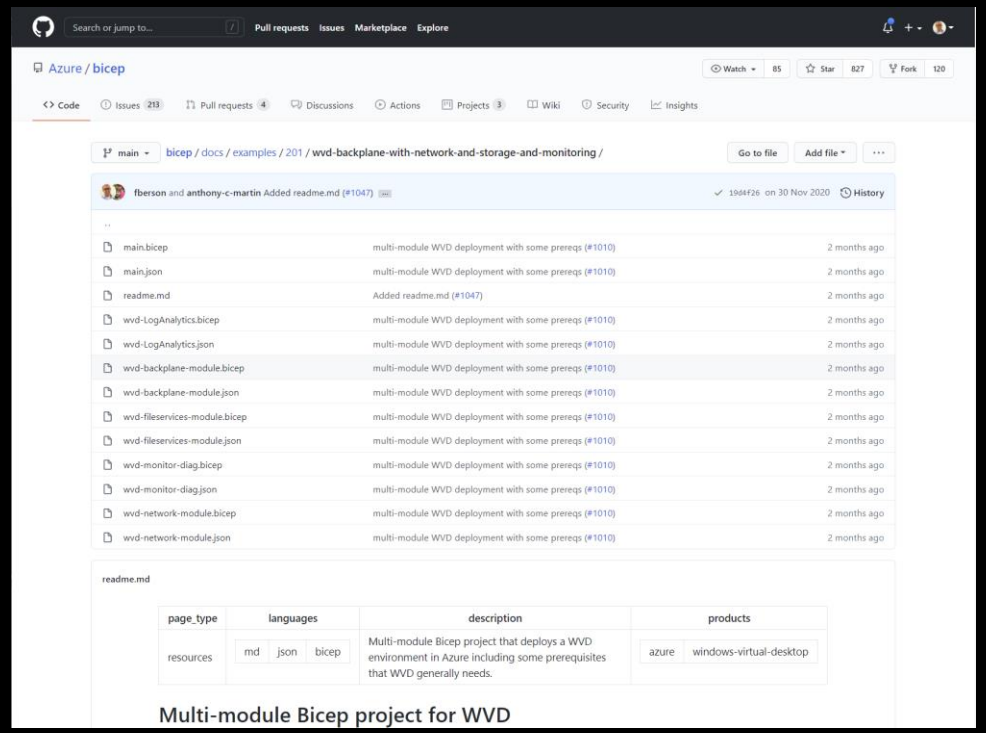
```
param adminUsername string

...

output username string = adminUsername
```

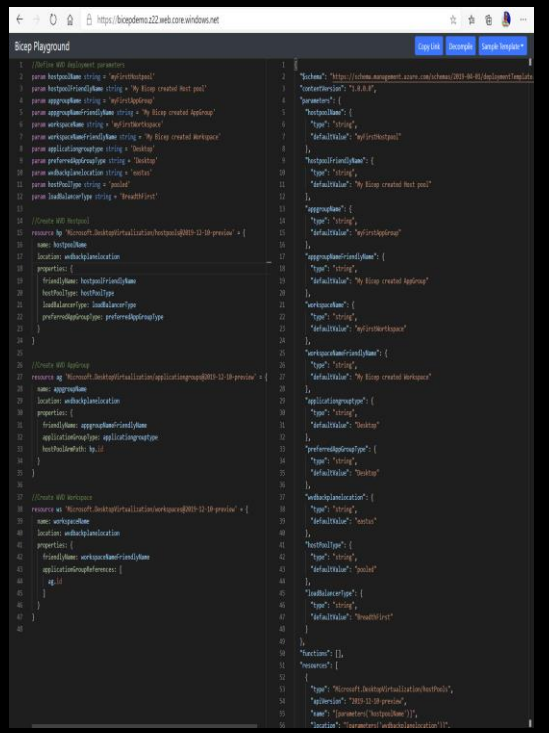
```
param adminUsername string

...

output adminUsername as string
```

Call to action: aka.ms/bicep

Install guides, tutorials, example code & playgrounds!







Esther Barthel

@virtuEs_IT

github.com/cognitionit

Freek Berson

@fberson

github.com/fberson