Flexing your Infrastructure as Code muscles with Bicep Templates!

US
Azure Virtual Desktop
User Group

Esther Barthel

@virtuEs_IT

github.com/cognitionit

Microsoft MVP

Freek Berson

@fberson

github.com/fberson

Microsoft MVP

ROADSH W
June 24th
May 20th
April 29th
April 14th
April 7th
February 17th
February 4th
January 29th
Azure Meetup Oslo
Belgium WVD User Group
Technine User Group
Detron Knowledge event
Dutch DevOps & Github comm.
Ask Wortell Live Stream
Azure Thursday
Azure Lowlands

BE A
NINJA
UNICORN

Agenda

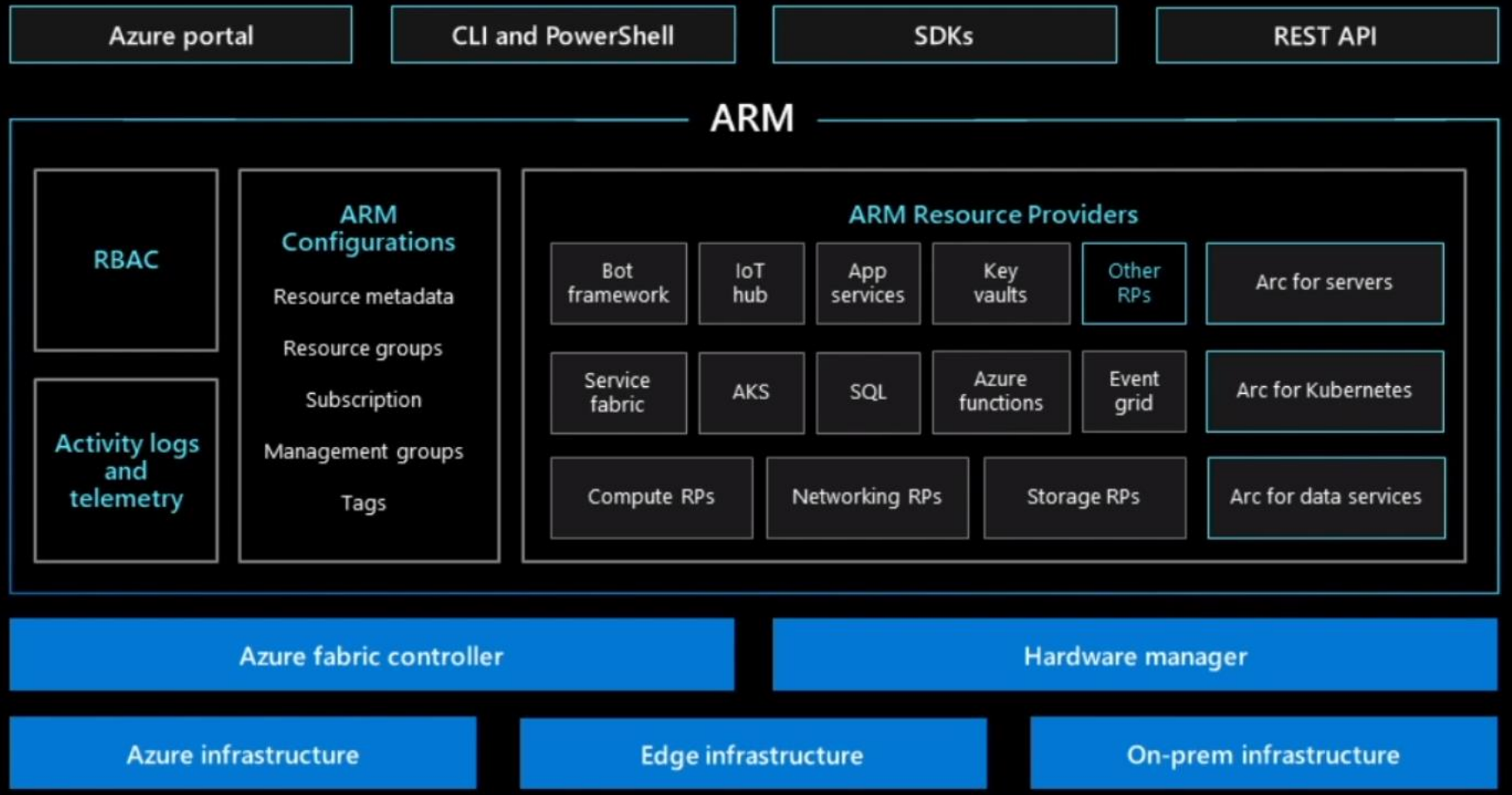
Azure Resource Manager & JSON

‘Bicep’ architecture

Bicep demos!

Bicep Roadmap & call to actions

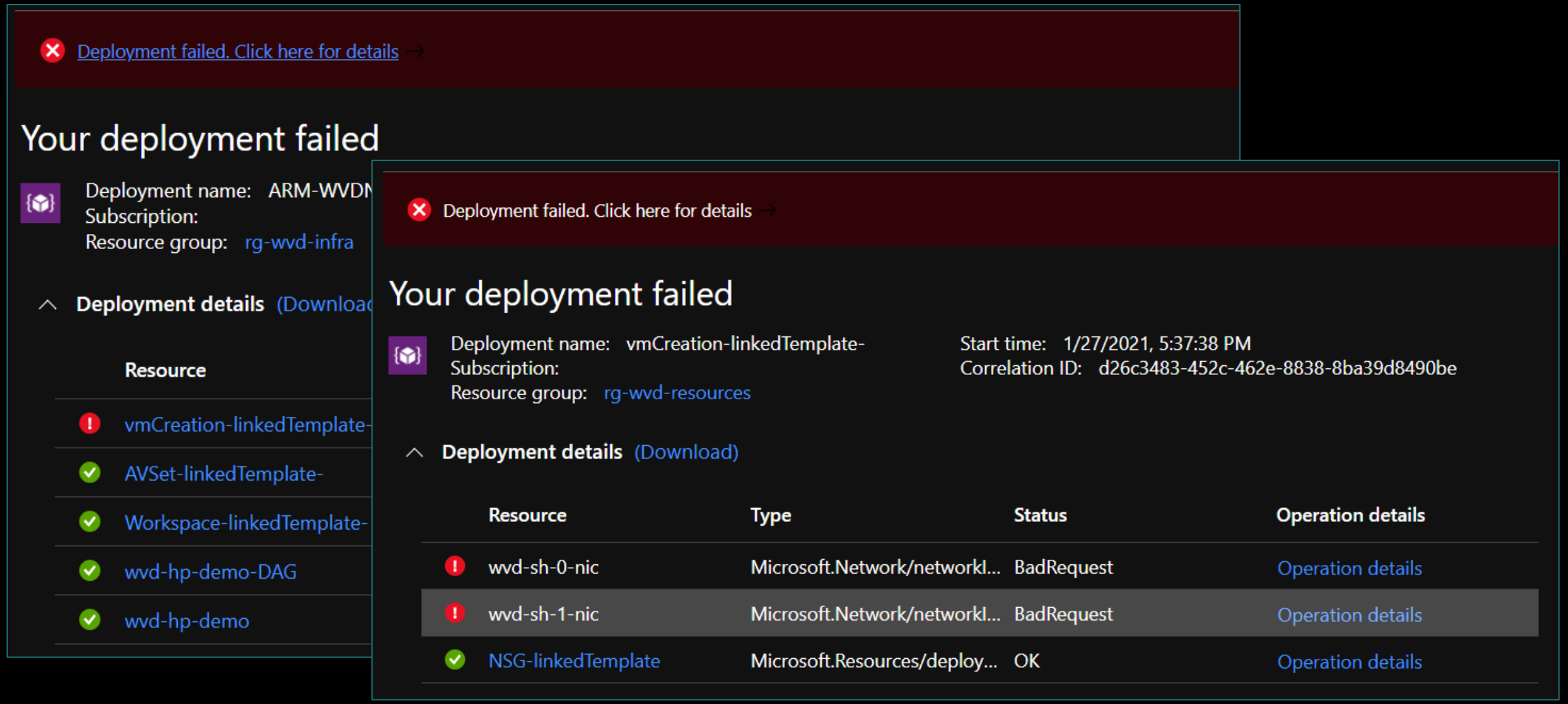
Azure Resource Manager



ARM Template – Resources

- **ARM template documentation**
  https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/
- **Reference guide:**
  https://docs.microsoft.com/en-us/azure/templates/
- **Learning Path:**
  https://docs.microsoft.com/en-us/learn/paths/deploy-manage-resource-manager-templates/
- **Azure Quickstart Templates:**
  https://azure.microsoft.com/en-us/resources/templates/

ARM Template – Automatic deployments



ARM Template – Automatic deployments

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    Select or create a parameter file to enable full validation...
    "parameters": {···
    },
    "variables": {···
    },
    "resources": [
        {
            "apiVersion": "2018-05-01",
            "name": "[concat('UpdateHostPool-', parameters('deploymentId'))]",
            "type": "Microsoft.Resources/deployments",
            "resourceGroup": "[parameters('hostpoolResourceGroup')]",
            "condition": "[not(empty(parameters('hostpoolResourceGroup')))]",
            "properties": {···
            }
        },
        {···
        },
        {
            "apiVersion": "2018-05-01",
            "name": "[concat('vmCreation-linkedTemplate-', parameters('deploymentId'))]",
            "resourceGroup": "[parameters('vmResourceGroup')]",
            "dependsOn": [
                "[concat('AVSet-linkedTemplate-', parameters('deploymentId'))]"
            ],
            "type": "Microsoft.Resources/deployments",
            "properties": {···
            }
        }
    ],
    "outputs": {
        "rdshVmNamesObject": {
            "value": "[variables('rdshVmNamesOutput')]",
            "type": "object"
        }
    }
}
```

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

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.

'Bicep'

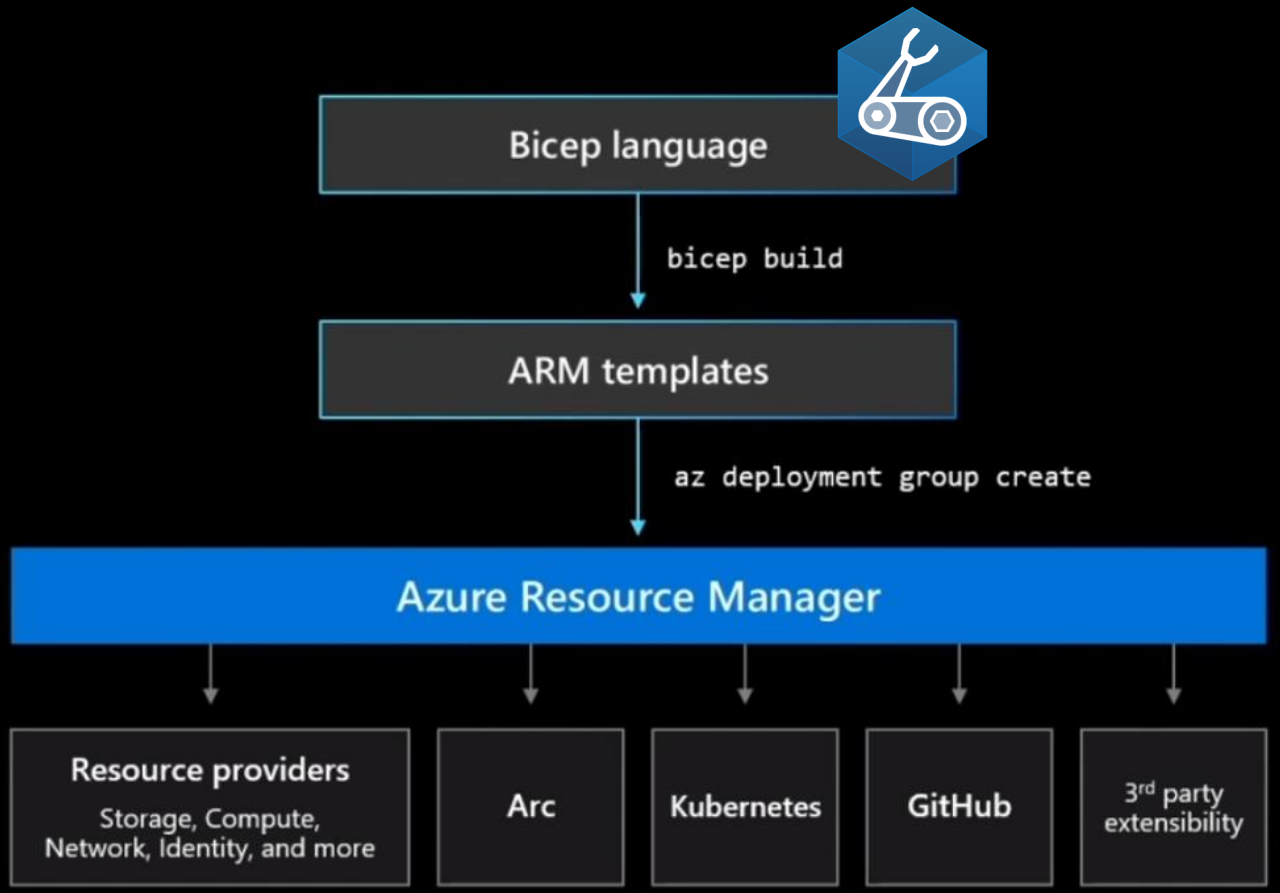
Simple declarative language to provision infrastructure to Azure.

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



Demo – Putting Bicep into Action!

Demo - Putting Bicep into GitHub Action

cognitionIT / AzureWVD

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

ARM deployment with bicep ARM deployment with bicep #28

Summary

Jobs

ARM Deployment with bicep file

ARM Deployment with bicep file
succeeded 6 minutes ago in 1m 44s

- Set up job
- Checkout
- Login via Az module
- Deploy ARM Resources with Bicep & Azure PowerShell
- Post Checkout
- Complete job

①

Deploy local 'Bicep' files

Note: Currently, both Azure CLI and Azure PowerShell can only deploy local Bicep files.
Bicep CLI is needed locally to compile Bicep files to JSON templates before deployment.

Azure CLI v2.20.0+

Azure CLI | Copy

```
az deployment group create \
  --name ExampleDeployment \
  --resource-group ExampleGroup \
  --template-file <path-to-template-or-bicep> \
  --parameters storageAccountType=Standard_GRS
```

Note: Azure CLI v2.20.0+ installs its own, self-contained version of Bicep CLI, which is not added to your PATH.

PowerShell 5.6.0+

Azure PowerShell

```
New-AzResourceGroupDeployment `
  -Name ExampleDeployment `
  -ResourceGroupName ExampleGroup `
  -TemplateFile <path-to-template-or-bicep> `
  -storageAccountType Standard_GRS
```

Note: Azure PowerShell does not have the capability to install the Bicep CLI yet. Azure PowerShell (v5.6.0+) expects that the Bicep CLI is already installed and available on the PATH.

Tip: run Bicep on windows-latest agent

Microsoft Windows Server 2019 Datacenter

- OS Version: 10.0.17763 Build 2061
- Image Version: 20210803.1

Installed Software

Language and Runtime

- Bash 4.4.23(1)-release
- Go 1.15.14
- Julia 1.6.2
- Node 14.17.4
- Perl 5.32.1
- PHP 8.0.9
- Python 3.7.9
- Ruby 2.5.9p229

Package Management

- Chocolatey 0.10.15
- NPM 6.14.14
- NuGet 5.10.0.7240

Tools

- 7zip 19.00
- azcopy 10.11.0
- Bicep 0.4.451
- Docker 20.10.6
- Docker-compose 1.29.2
- Git 2.32.0
- OpenSSL 1.1.1
- Packer 1.7.4

CLI Tools

- Azure CLI 2.27.0
- Azure DevOps CLI extension 0.18.0
- Cloud Foundry CLI 6.53.0
- GitHub CLI 1.13.1

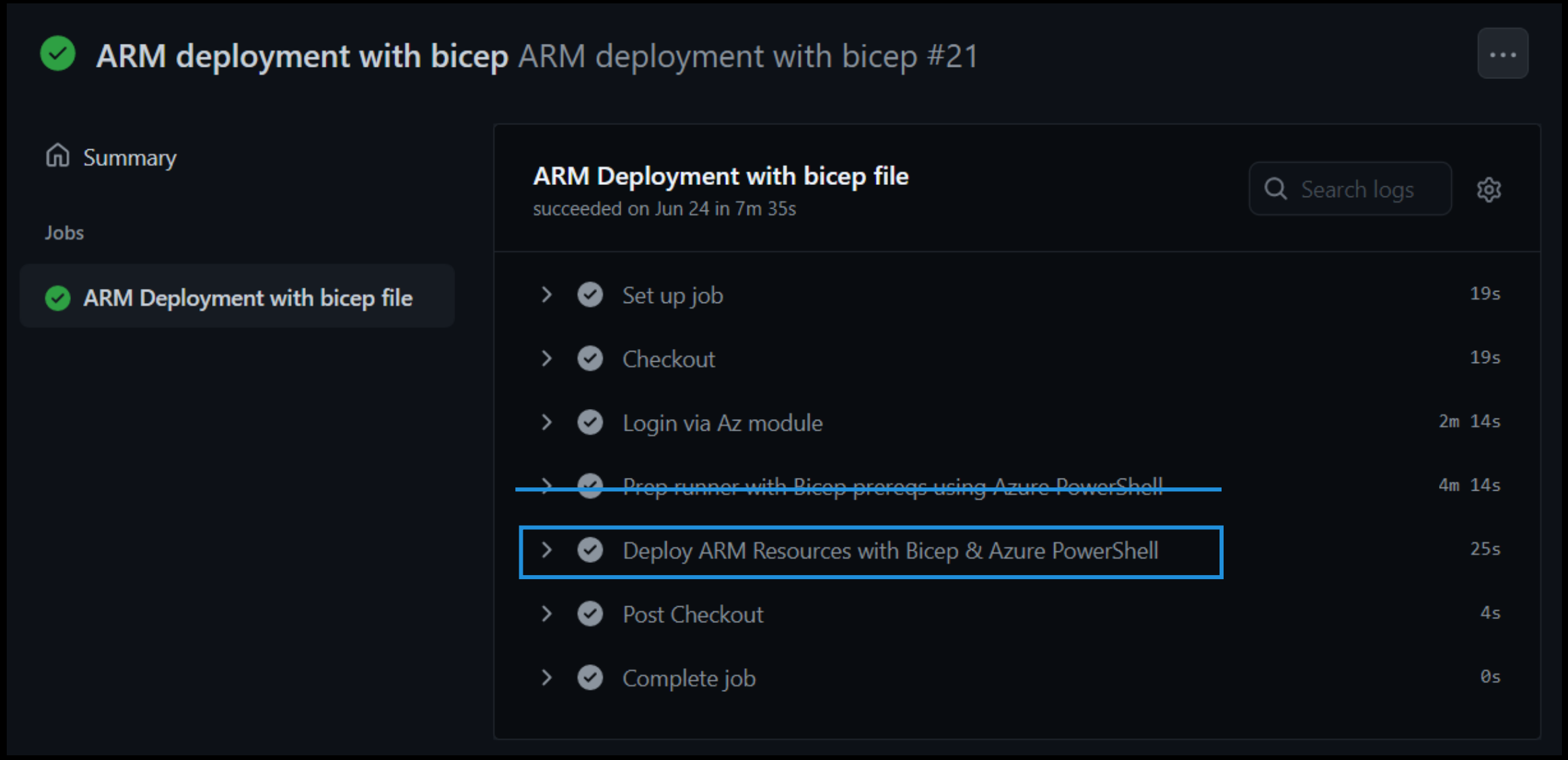
PowerShell Tools

- PowerShell 7.1.3

Azure Powershell Modules

Module	Version
Az	6.1.0
Azure	5.3.0
AzureRM	6.13.1

Putting Bicep into GitHub Action



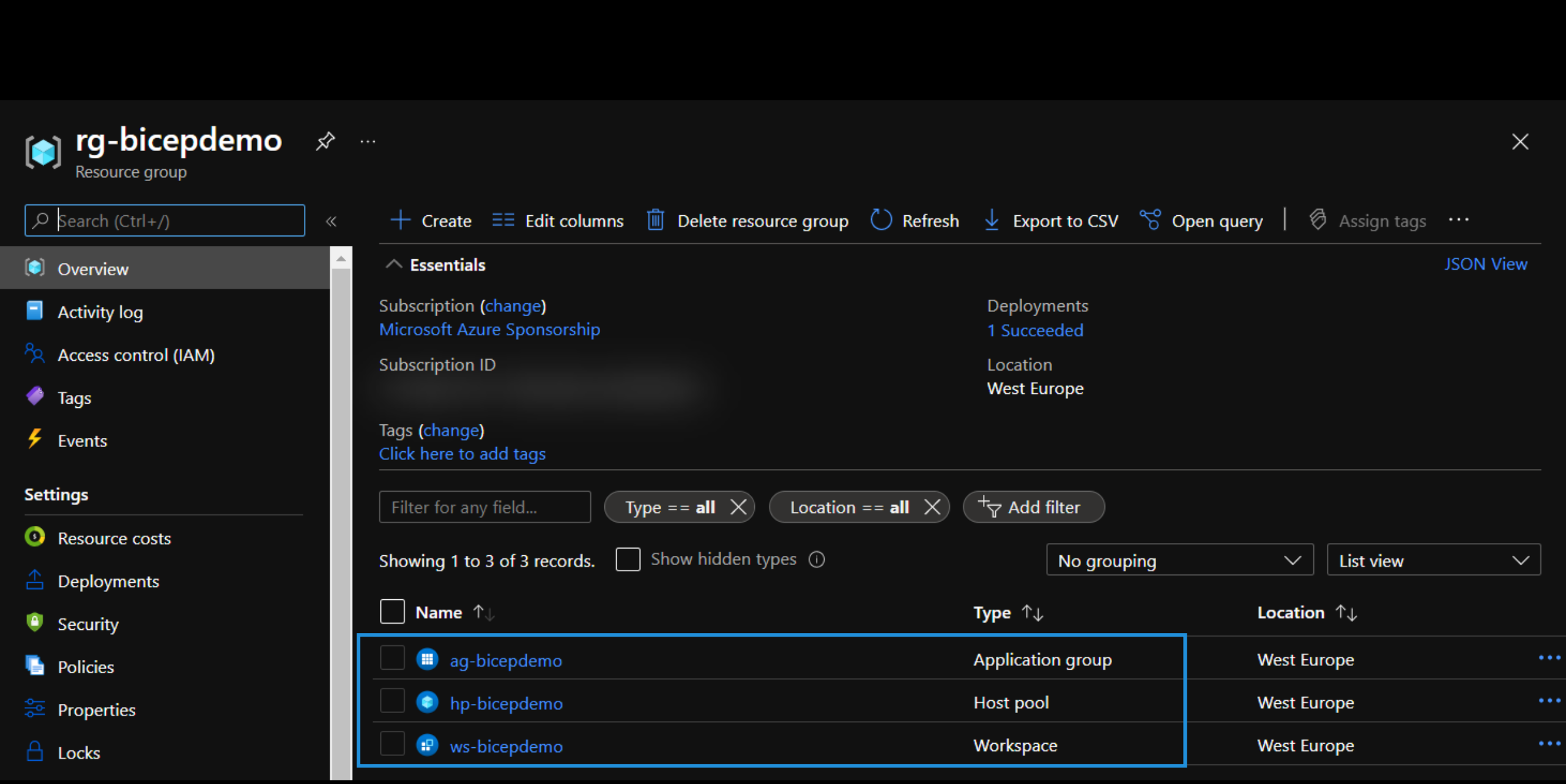
```yaml
# Action = Azure PowerShell: Run inline script
# source: https://github.com/marketplace/actions/azure-powershell-action
- name: Deploy ARM Resources with Bicep & Azure PowerShell
  uses: azure/powershell@v1
  with:
    inlineScript: |
      # Read the GitHub Actions variables
      [string]$githubWorkspace = "${{GITHUB.WORKSPACE}}"
      # Read the environment variables in PowerShell
      [string]$location = [System.Environment]::GetEnvironmentVariable('LOCATION')
      [string]$bicepFile = [System.Environment]::GetEnvironmentVariable('BICEP_FILE')
      [string]$resourcegroupName = [System.Environment]::GetEnvironmentVariable('RESOURCE_GROUP_NAME')

      $namePostFix = $resourcegroupName.Replace("rg-","")

      ## Create a Template Parameter Object (hashtable)
      $objTemplateParameter = @{
        "location" = "$($location)";
        "workSpaceName" = "ws-$($namePostFix)";
        "hostpoolName" = "hp-$($namePostFix)";
        "appgroupName" = "ag-$($namePostFix)";
        "preferredAppGroupType" = "Desktop";
        "hostPoolType" = "pooled";
        "loadbalancertype" = "DepthFirst";
        "appgroupType" = "Desktop";
      }
      # Location of the bicep file in the local checked-out repo
      $biceptemplateFile = [string]("$($githubWorkspace)" + "\bicep\" + "$($bicepFile)")

      ## Deploy resources based on bicep file for ARM Template
      New-AzResourceGroupDeployment -ResourceGroupName $resourcegroupName -TemplateFile $($biceptemplateFile) -TemplateParameterObject $objTemplateParameter
    azPSVersion: 'latest'
```

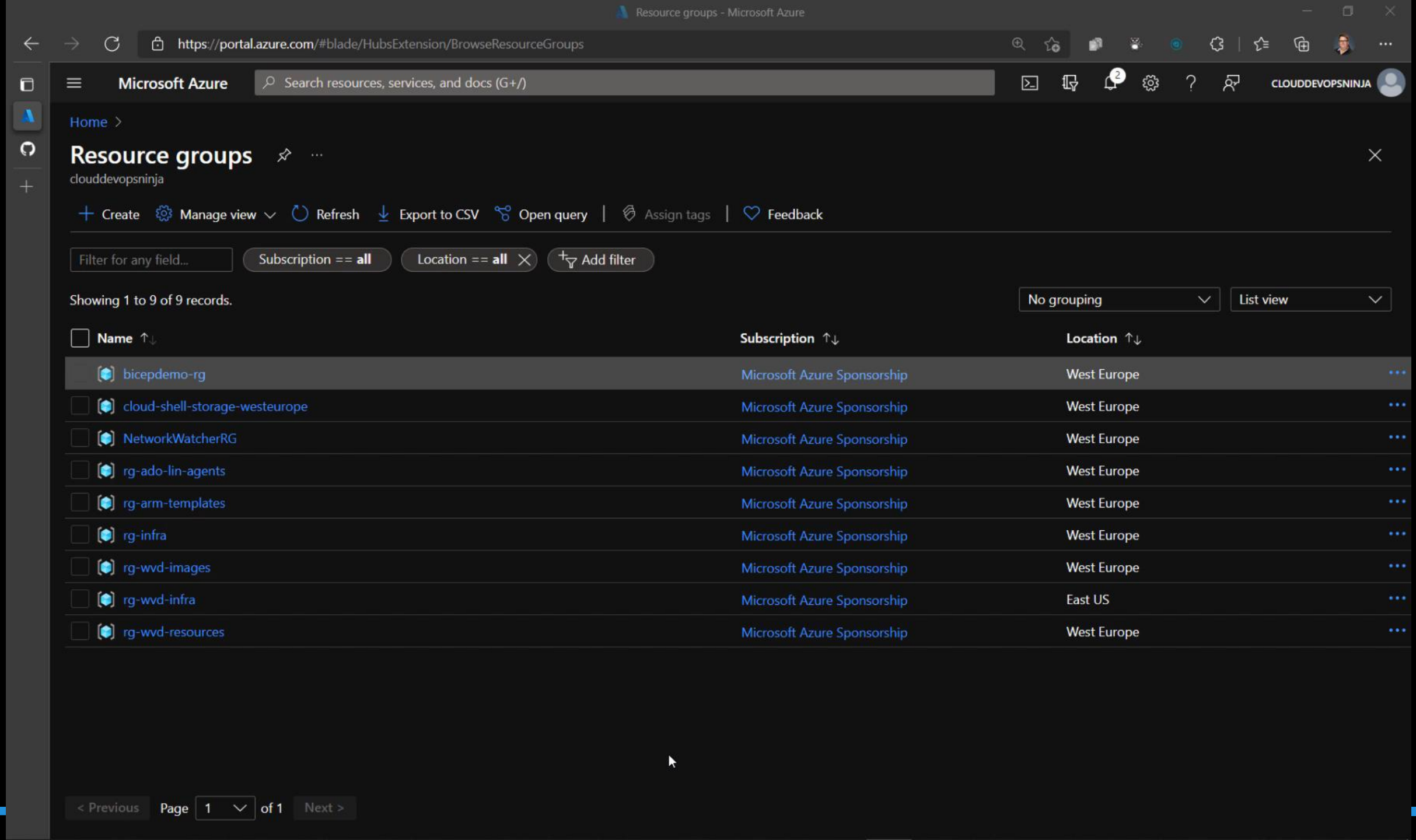
Putting Bicep into GitHub Action

ARM Deployment with bicep file
succeeded on Mar 16 in 6m 54s

- Set up job — 24s
- Checkout — 14s
- Login via Az module — 1m 54s
- Deploy ARM Resources with Bicep & Azure PowerShell — 23s

```
74  * BICEP FILE: demo.bicep
75  * RESOURCE GROUP NAME: rg-bicepdemo
76  * GITHUB_WORKSPACE: D:\a\AzureWVD\AzureWVD
77  * BICEP TEMPLATE FILE: D:\a\AzureWVD\AzureWVD\bicep\demo.bicep
85  VERBOSE: Performing the operation "Creating Deployment" on target "rg-bicepdemo".
86  VERBOSE: 9:35:24 PM - Template is valid.
87  VERBOSE: 9:35:26 PM - Create template deployment 'demo'
89  VERBOSE: 9:35:31 PM - Resource Microsoft.DesktopVirtualization/hostPools 'hp-bicepdemo' provisioning status is succeeded
91  VERBOSE: 9:35:37 PM - Resource Microsoft.DesktopVirtualization/workspaces 'ws-bicepdemo' provisioning status is succeeded
92  VERBOSE: 9:35:37 PM - Resource Microsoft.DesktopVirtualization/applicationGroups 'ag-bicepdemo' provisioning status is succeeded
94  ResourceGroupName       : rg-bicepdemo
95  OnErrorDeployment       :
96  DeploymentName          : demo
97  CorrelationId           : 3f74503c-3013-47e1-a307-854615ee81a2
98  ProvisioningState       : Succeeded
99  Timestamp               : 3/16/2021 9:35:36 PM
100 Mode                    : Incremental
```


rg-bicepdemo
Resource group
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Events
Settings
Resource costs
Deployments
Security
Policies
Properties
Locks
Create
Edit columns
Delete resource group
Refresh
Export to CSV
Open query
Assign tags
Essentials
JSON View
Subscription (change)
Microsoft Azure Sponsorship
Subscription ID
Deployments
1 Succeeded
Location
West Europe
Tags (change)
Click here to add tags
Filter for any field...
Type == all
Location == all
Add filter
Showing 1 to 3 of 3 records.
Show hidden types
No grouping
List view
Name
Type
Location
ag-bicepdemo
Application group
West Europe
hp-bicepdemo
Host pool
West Europe
ws-bicepdemo
Workspace
West Europe

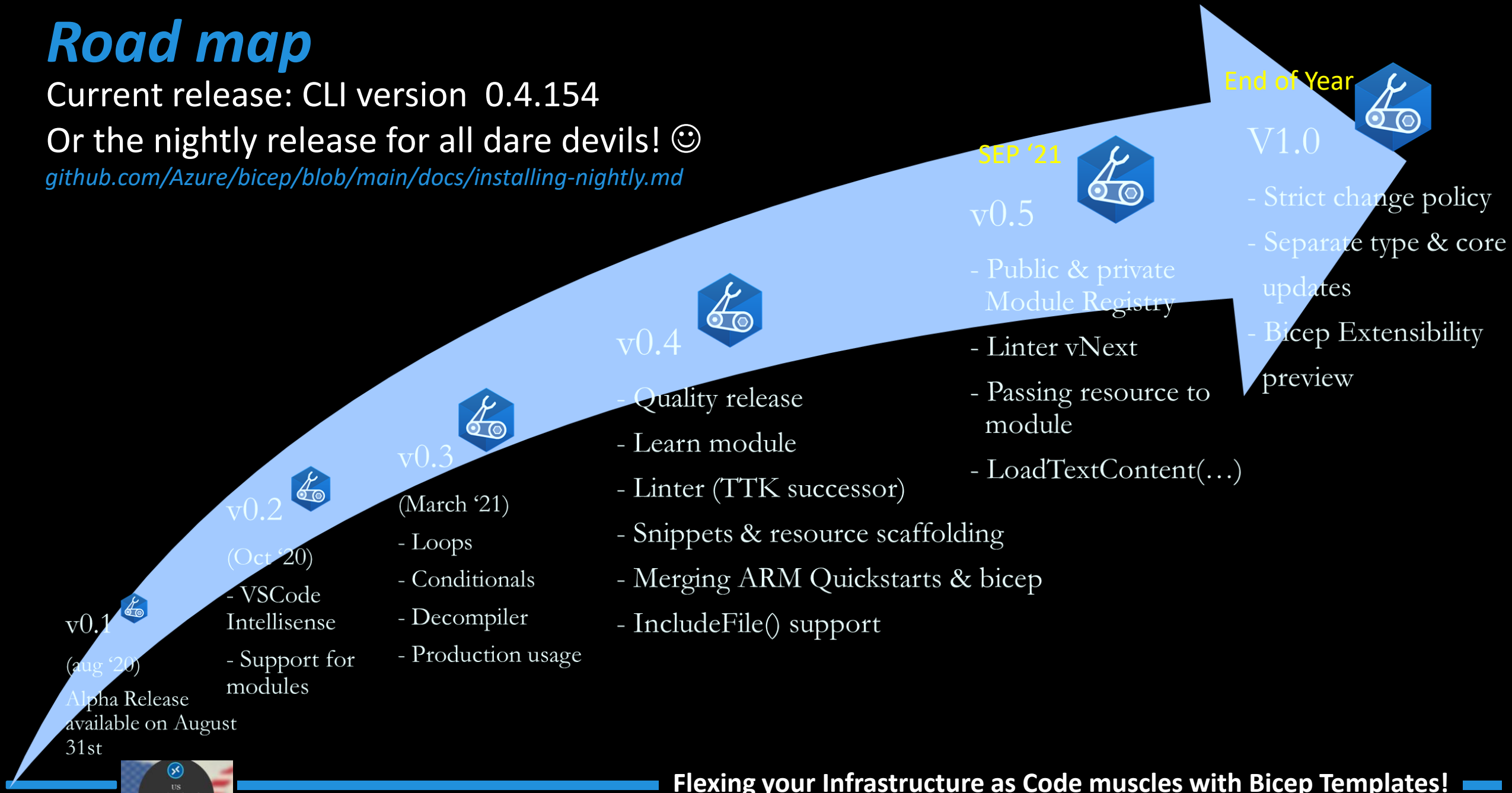

Resource groups - Microsoft Azure
https://portal.azure.com/#blade/HubsExtension/BrowseResourceGroups
Microsoft Azure
Search resources, services, and docs (G+/)
CLOUDDEVOPSNINJA
Home
Resource groups
clouddevopsninja
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Feedback
Filter for any field...
Subscription == all
Location == all
Add filter
Showing 1 to 9 of 9 records.
No grouping
List view
Name
Subscription
Location
bicepdemo-rg
Microsoft Azure Sponsorship
West Europe
cloud-shell-storage-westeurope
Microsoft Azure Sponsorship
West Europe
NetworkWatcherRG
Microsoft Azure Sponsorship
West Europe
rg-ado-lin-agents
Microsoft Azure Sponsorship
West Europe
rg-arm-templates
Microsoft Azure Sponsorship
West Europe
rg-infra
Microsoft Azure Sponsorship
West Europe
rg-wvd-images
Microsoft Azure Sponsorship
West Europe
rg-wvd-infra
Microsoft Azure Sponsorship
East US
rg-wvd-resources
Microsoft Azure Sponsorship
West Europe
< Previous
Page 1 of 1
Next >

Road map

Current release: CLI version 0.4.154

Or the nightly release for all dare devils! ☺

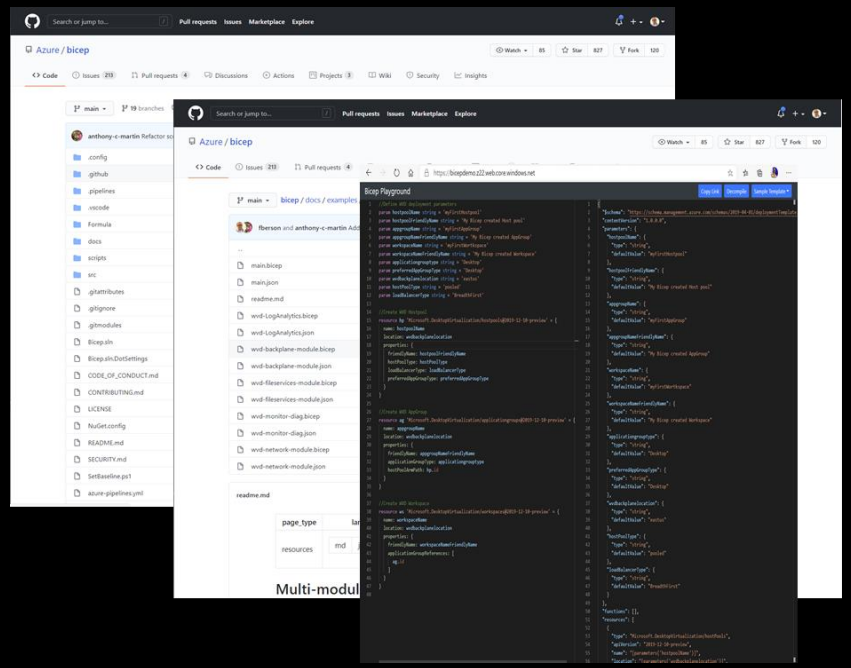
github.com/Azure/bicep/blob/main/docs/installing-nightly.md



Call to actions:

Install guides, tutorials, example code & playgrounds!

aka.ms/bicep



Aka.ms/learnbicep

Introductory path

The Deploy and manage resources in Azure by using Bicep learning path is the best place to start. It introduces you to the concept of infrastructure as code. The path takes you through the steps of building increasingly complex Bicep files.

This path contains the following modules.

Learn module	Description
Introduction to infrastructure as code using Bicep	This module describes the benefits of Manager, and Bicep to quickly and cor helps you determine the types of depl deployment tool.
Build your first Bicep template	In this module, you define Azure resou the consistency and reliability of your required, and scale your deployments flexible and reusable by using paramet
Build reusable Bicep templates by using parameters	This module describes how you can us for your template during each deploym decorators, which make your paramete also learn about the different ways tha protect them when you're working wit
Build flexible Bicep templates by using conditions and loops	Learn how to use conditions to deploy are in place. Also learn how to use loo similar properties.
Deploy child and extension resources by using Bicep	This module shows how to deploy vari Learn about child and extension resou within Bicep. Use Bicep to work with re template or module.
Deploy resources to subscriptions, management groups, and tenants by using Bicep	Deploy Azure resources at the subscrip scope. Learn what these resources are, create Bicep code to deploy them. Also files that you can deploy across multip
Extend templates by using deployment scripts	Learn how to add custom steps to you template (ARM template) by using dep

Other modules

In addition to the preceding path, the following modules contain Bicep content.

Learn module	Description
Manage changes to your Bicep code by using Git	Learn how to use Git to support your Bicep development workflow by keeping track of the changes you make as you work. You'll find out how to commit files, view the history of the files you've changed, and how to use branches to develop multiple versions of your code at the same time. You'll also learn how to use GitHub or Azure Repos to publish a repository so that you can collaborate with team members.
Publish libraries of reusable infrastructure code by using template specs	Template specs enable you to reuse and share your ARM templates across your organization. Learn how to create and publish template specs, and how to deploy them. You'll also learn how to manage template specs, including how to control access and how to safely update them by using versions.
Preview Azure deployment changes by using what-if	This module teaches you how to preview your changes with the what-if operation. By using what-if, you can make sure your Bicep file only makes changes that you expect.
Authenticate your Azure deployment pipeline by using service principals	Service principals enable your deployment pipelines to authenticate securely with Azure. In this module, you'll learn what service principals are, how they work, and how to create them. You'll also learn how to grant them permission to your Azure resources so that your pipelines can deploy your Bicep files.

Esther Barthel

@virtuEs_IT

github.com/cognitionit

Freek Berson

@fberson

github.com/fberson