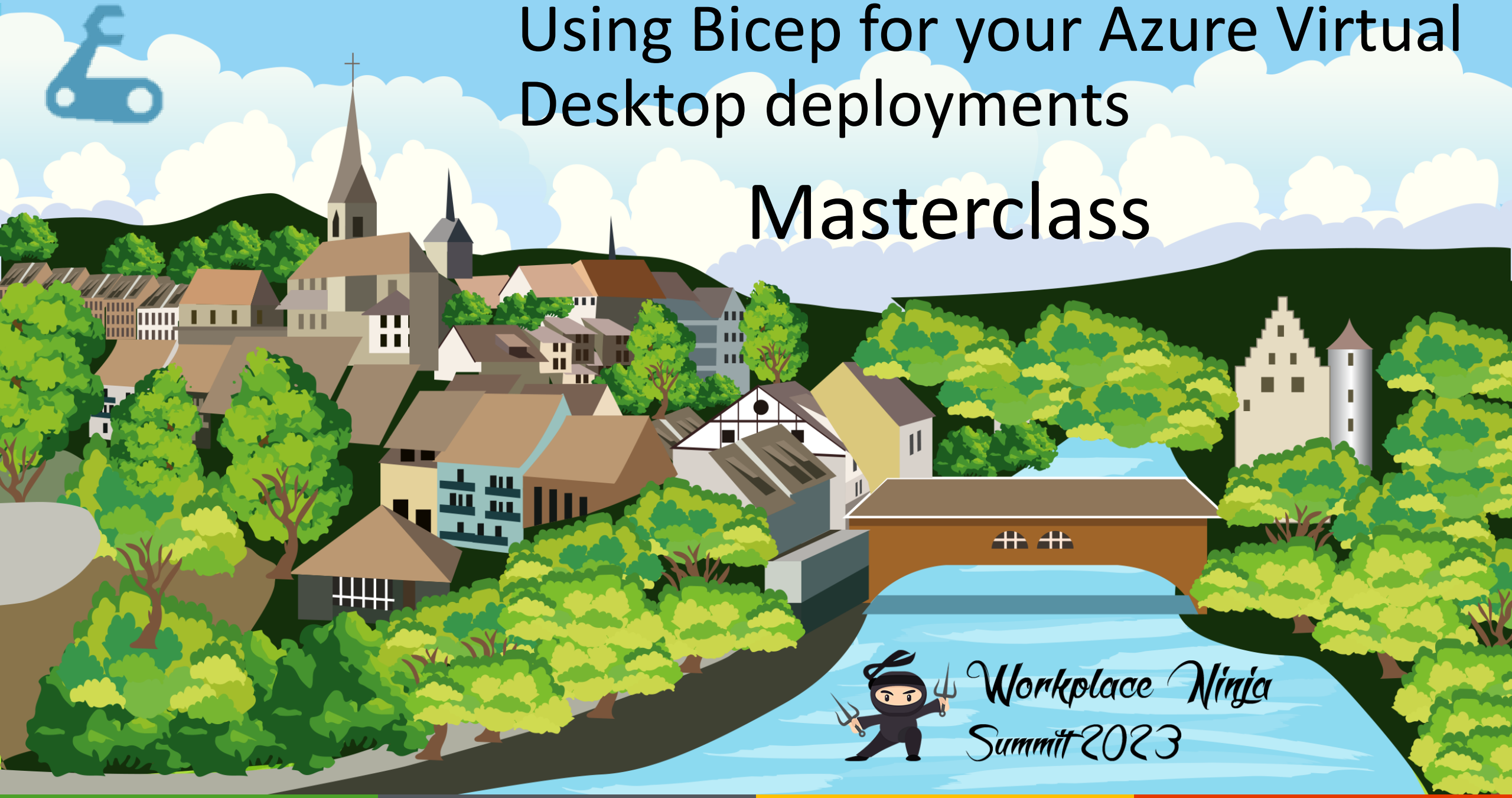
Using Bicep for your Azure Virtual Desktop deployments

Masterclass



www.wpninjas.eu
#WPNinjaS

Diamond Sponsor

Microsoft

Platinum Sponsors

2Pint

PATCH MY PC

glueckkanja gab

Gold Sponsors

RECAST SOFTWARE

baseVISION
SECURE & MODERN ENDPOINT MANAGEMENT

LIQUIT

controlUP

Rimo3

adaptiva

Silver Sponsors

APENTO

sepago
a PROACT company

EPIC FUSION
BRING IT ALL TOGETHER

onevinn
number one in intelligent security

About Esther Barthel

www.wpninjas.eu
#WPNinjaS

Focus

Cloud DevOps Engineer

From

The Netherlands

My Blog

https://www.cloud-devops.ninja/

Certifications

Microsoft Certified Azure Virtual Desktop Specialty

Microsoft MVP Most Valuable Professional

Citrix Technology Professional

Hobbies

KAMADOJOE

MARVEL STUDIOS

Contact

@virtuEs_IT

github.com/cloud-devops-ninja

About Freek Berson

Focus

End User Computing, Infrastructure as Code

From

The Netherlands

My Blog

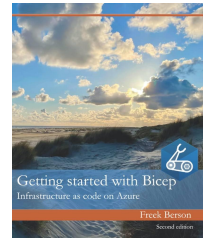
Themicrosoftplatform.net

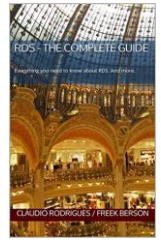
Certifications

Microsoft® Most Valuable Professional

Books

- Getting started with Bicep
- RDS - The Complete Guide





Contact

@fberson

github.com/fberson

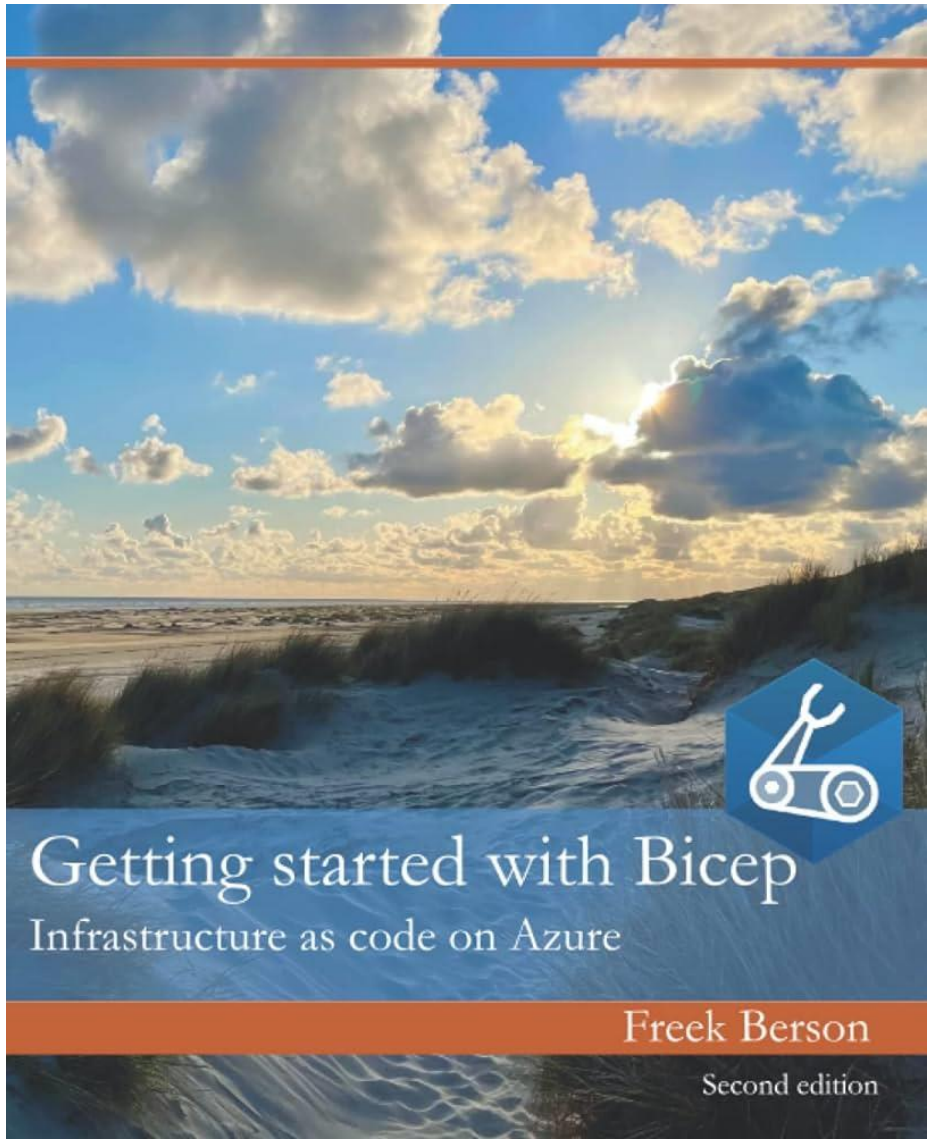
Tweet to win!

Tweet about this session and win the book!

@fberson

@virtuEs_IT

#WPNinjaS



Agenda

Recap: What is Bicep (in 300 seconds)

For those who are new and did not join us yesterday

Demo 1: Advanced topics

Full AVD, Modules, Module registries, deployment Stacks, Graph API

Demo 2: Deploying to Azure

Deploy using Azure DevOps, CLI, PowerShell, and other tips & ticks

Q&A

Key takeaways:

- **Get familiar with advanced Bicep topics**
- **Learn how to deploy to Azure with DevOps, CLI, and more!**

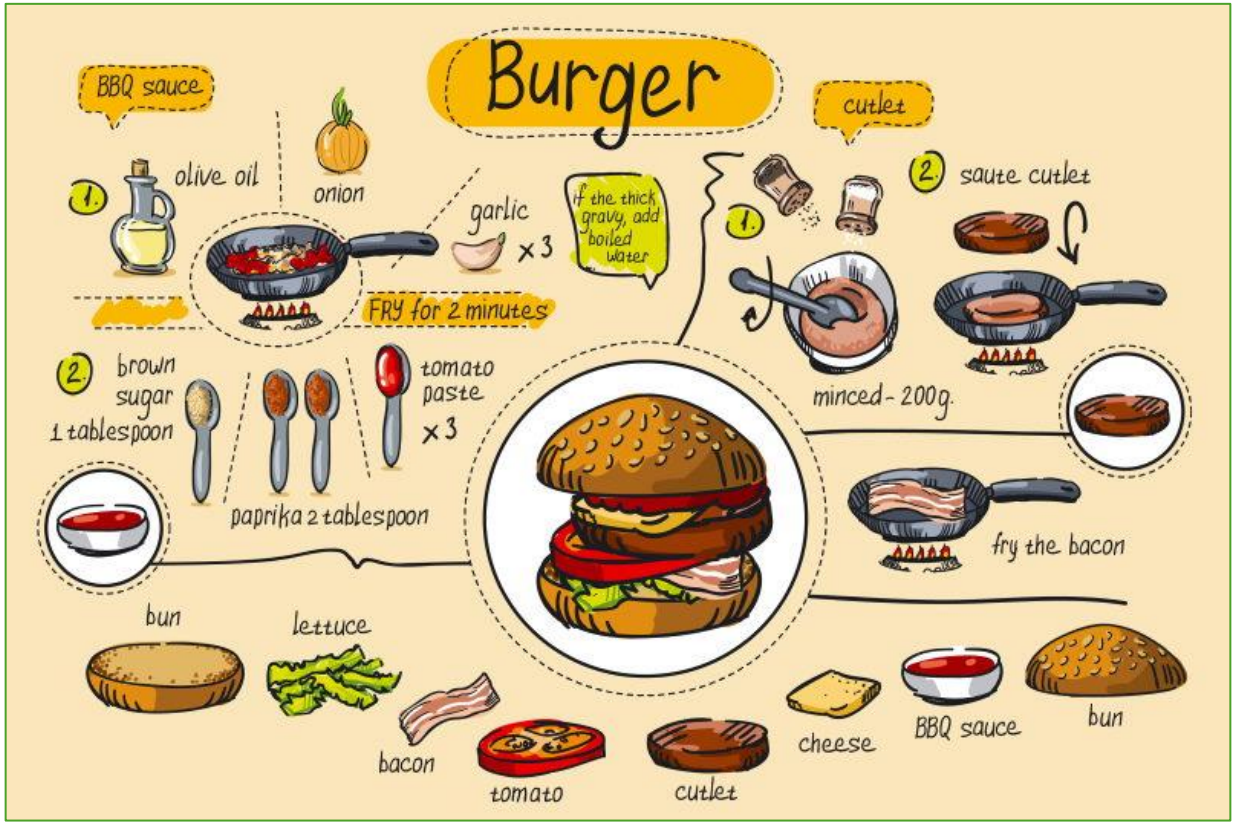
Recap: What is Bicep?

… in 300 seconds!

Imperative vs. Declarative

execute a sequence of commands

Like a **step-by-step instruction** manual.



specify only the end configuration

Like **ordering from a menu.**

Imperative vs. Declarative

Scripts

```powershell
# 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
```

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'
  }
}
```

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",
```

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."*

Where to position Bicep?

Intuitive

Easy to read and to author

Transpiles to ARM Templates

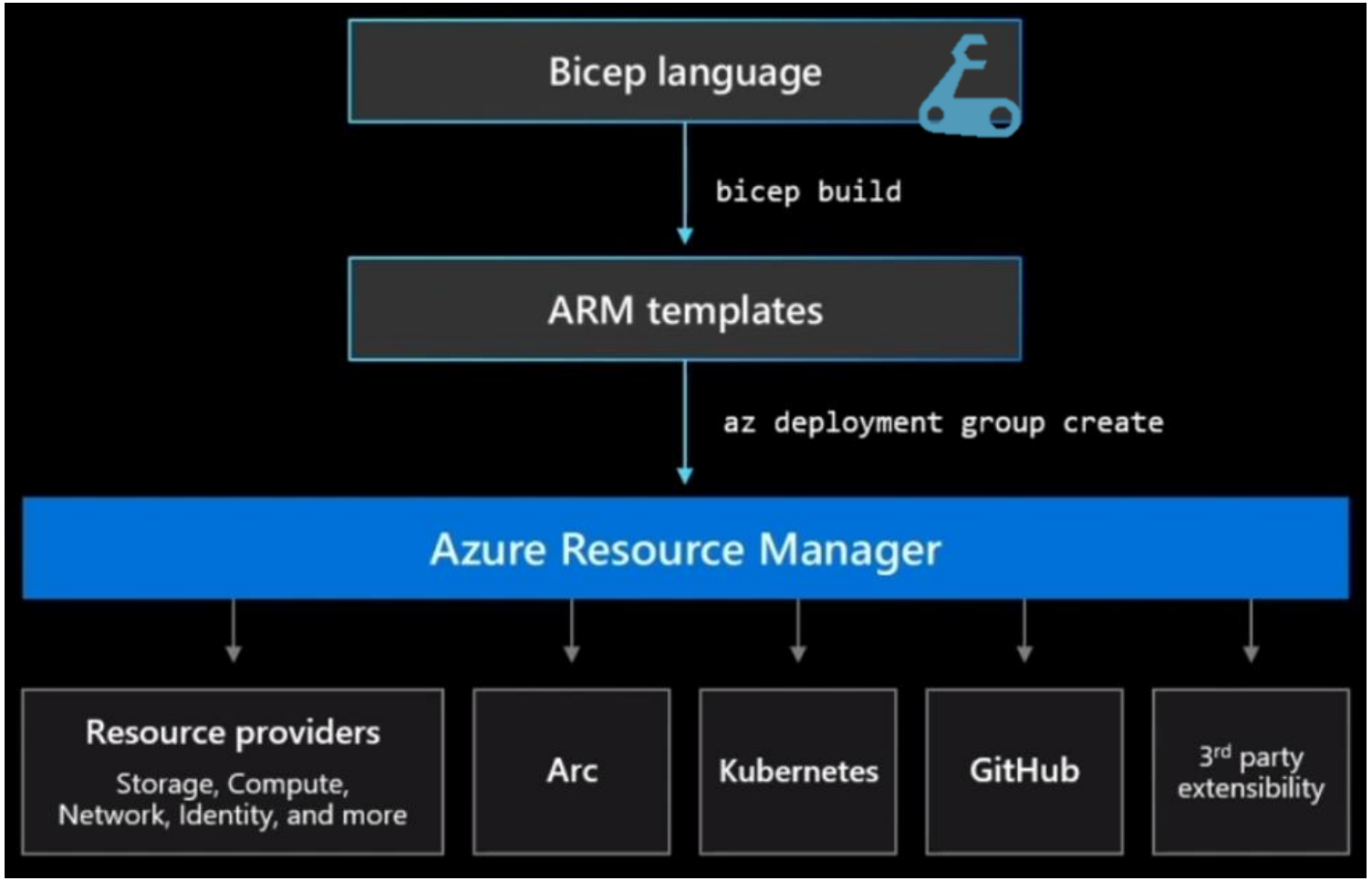
Leverage ARM template knowledge and investments

Modular

Abstract common blocks of code into reusable parts

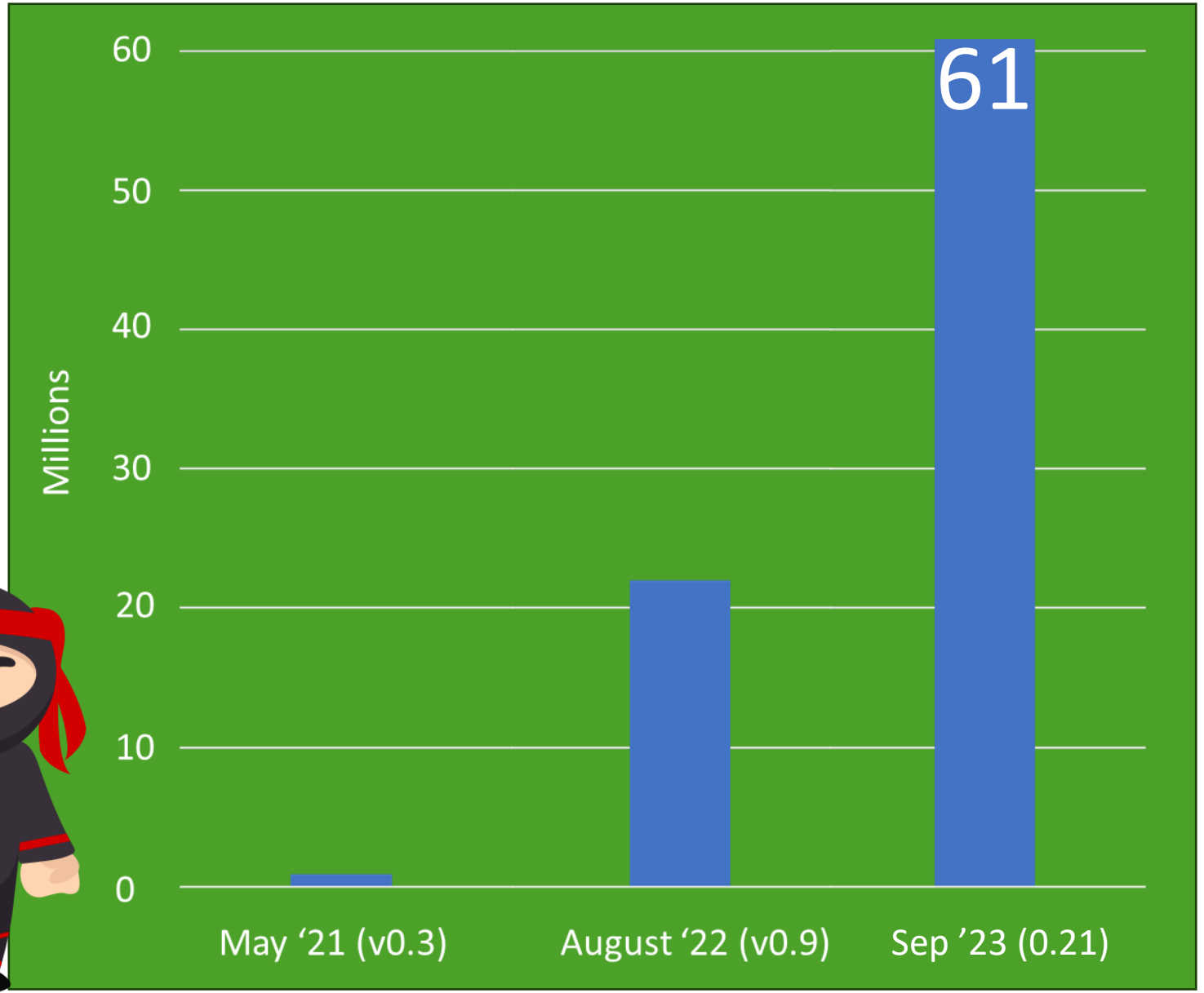
Open Source

Transparency and community



How well is Bicep adopted?

Number of resources deployed using Bicep in last 30 days



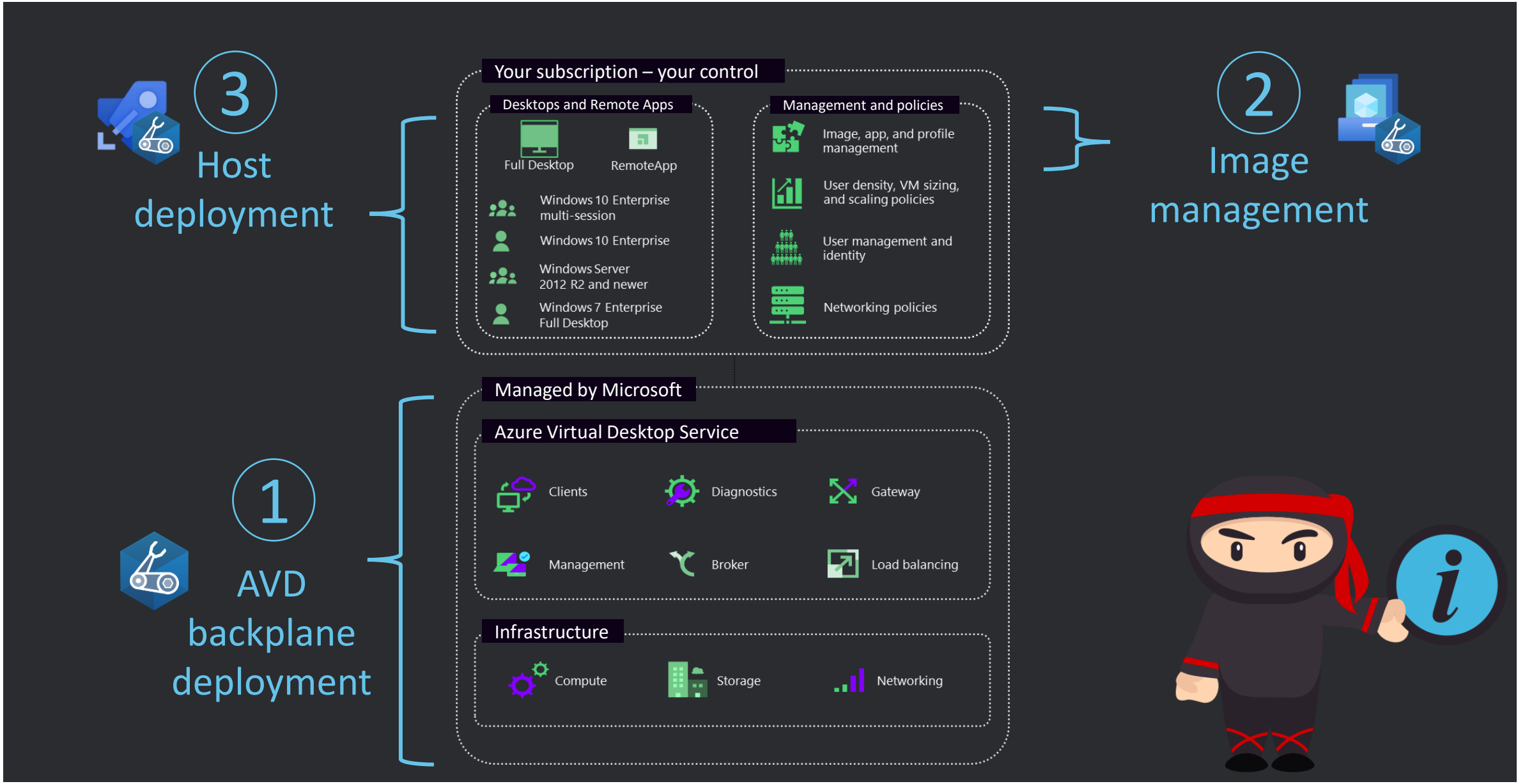
Demo time!

Advanced Bicep examples &
a full AVD deployment!

Demo time!

Deploying using DevOps, CLI, and PowerShell

Putting Bicep into (IaC) action



Getting started with Bicep!

1. Install the tools (bicep.exe, Visual Studio Code)

https://aka.ms/bicep

+

Visual Studio Code

Infra-as-Code with Bicep!

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

https://aka.ms/bicep

Deploy Bicep using az CLI

deployCLI.ps1

```powershell
# prerequisite: local install Azure CLI (2.20.0 or later)
az version
# prerequisite: connect to Azure (interactive)
az login
# prerequisite: set a subscription to be the current active subscription
az account set --name 'Visual Studio Enterprise'

# deploy to resourcegroup, using az CLI
az deployment group create --resource-group '_rg-WPNS-demo' `
                           --template-file 'deployCLI.bicep' `
                           --parameters hostpoolName='hp-demo-wpns-cli'

```

Deploy Bicep using az CLI

Deploy Bicep using Az PowerShell Module

deployPS.ps1 > ...

```powershell
# prerequisite: local install Azure PowerShell Module (5.6.0 or later)
Get-Module -Name Az -ListAvailable
# prerequisite: local install Bicep CLI
bicep.exe --version
# prerequisite: connect to Azure (interactive)
Connect-AzAccount | Out-Null
# prerequisite: set a subscription to be the current active subscription
$subscription = Get-AzSubscription -SubscriptionName 'Visual Studio Enterprise'
Select-AzSubscription -SubscriptionObject $subscription | Out-Null

# deploy to resourcegroup, using Az PowerShell Module
$params = @{
    hostpoolName = 'hp-demo-wpns-ps'
}
New-AzResourceGroupDeployment -ResourceGroupName '_rg-WPNS-demo' -TemplateFile 'deployCLI.bicep' `
                              -TemplateParameterObject $params
```

Deploy Bicep using Az PowerShell Module


```
DeploymentName           : deployCLI
ResourceGroupName        : _rg-WPNS-demo
ProvisioningState        : Succeeded
Timestamp                : 2023-09-24 19:31:42
Mode                     : Incremental
TemplateLink             :
Parameters               :
                           Name                   Type                       Value
                           =====================  =========================  ==========
                           location               String                     "westeurope"
                           hostpoolName           String                     "hp-demo-WPNS-ps"
                           hostpoolType           String                     "Pooled"
                           loadBalancerType       String                     "BreadthFirst"
                           preferredAppGroupType  String                     "Desktop"
                           enableValidationMode   Bool                       false


Outputs                  :
DeploymentDebugLogLevel  :
```

Deploy Bicep using Azure DevOps Pipeline


```yaml
parameters:
  - name: resourcegroupName
    type: string
    default: "_rg-WPNS-demo"
  - name: hostpoolName
    type: string
    default: "hp-demo-wpns-ado"

stages:
  - stage: "AzureResourceDeployment"
    jobs:
      - job: "BicepTemplateDeployment"
        steps:
          # ARM template deployment v3 | Deploy a resource, using a Bicep template.
          - task: AzureResourceManagerTemplateDeployment@3
            inputs:
              # Azure Details
              deploymentScope: 'Resource Group'
              azureResourceManagerConnection: 'VisualStudioEnterprise'
              action: 'Create Or Update Resource Group'
              resourceGroupName: '${{parameters.resourcegroupName}}'
              location: 'westeurope'
              # Template
              templateLocation: 'Linked artifact'
              csmFile: '$(System.DefaultWorkingDirectory)\deployADO.bicep'
              overrideParameters: '-hostpoolName ${{parameters.hostpoolName}}'
              deploymentMode: "Incremental"
```

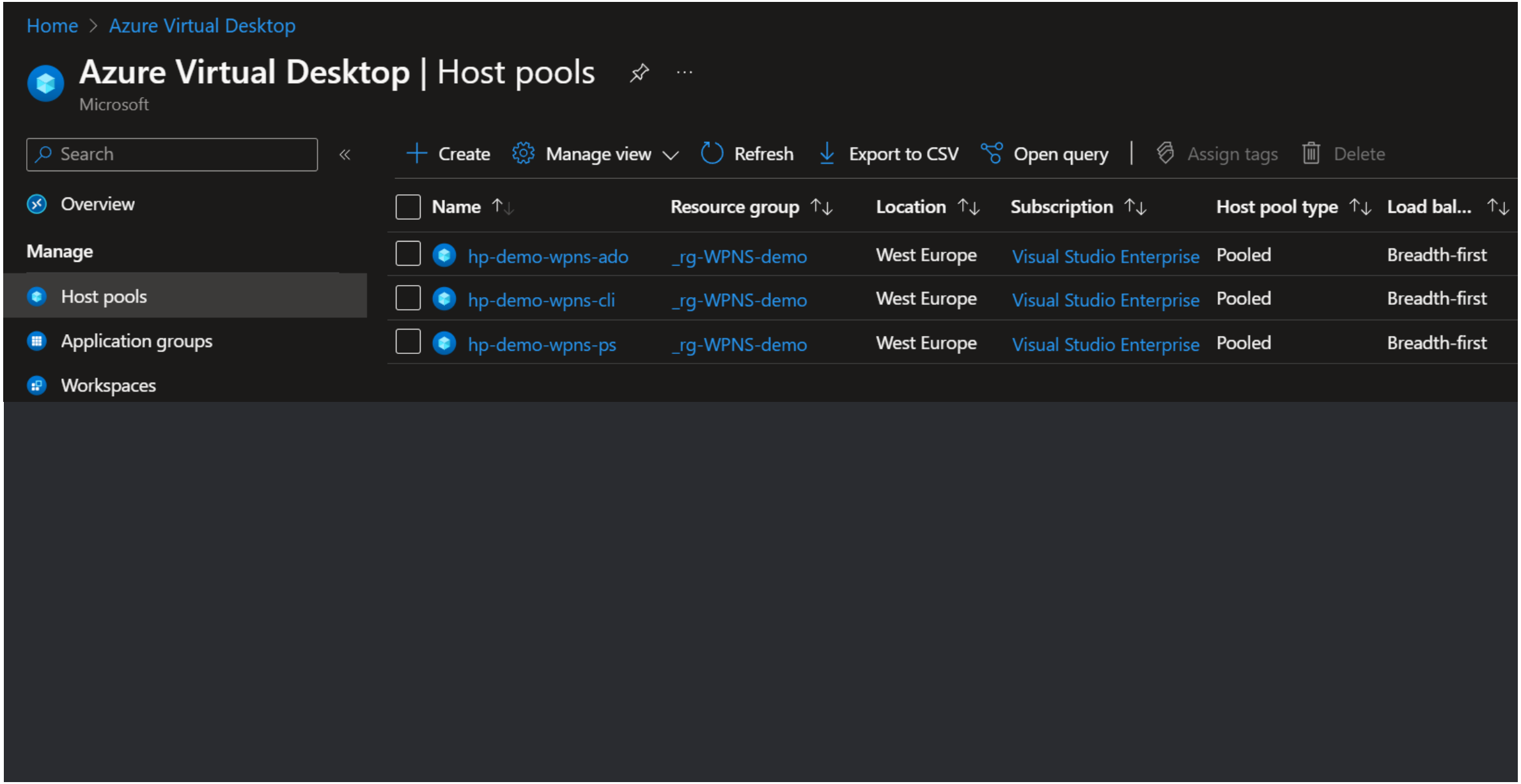
Deploy Bicep using Azure DevOps Pipeline

AzureResourceManagerTemplateDeployment

```
1  Starting: AzureResourceManagerTemplateDeployment
2  ==============================================================================
3  Task         : ARM template deployment
4  Description  : Deploy an Azure Resource Manager (ARM) template to all the deployment scopes
5  Version      : 3.227.0
6  Author       : Microsoft Corporation
7  Help         : https://docs.microsoft.com/azure/devops/pipelines/tasks/deploy/azure-resource-group-deployment
8  ==============================================================================
9  ARM Service Connection deployment scope - Subscription
10 Checking if the following resource group exists: _rg-WPNS-demo.
11 Resource group exists: true.
12 Creating deployment parameters.
13 Setting active cloud to: AzureCloud
38 Starting template validation.
39 Deployment name is deployADO-20230924-202539-7dae
40 Template deployment validation was completed successfully.
41 Starting Deployment.
42 Deployment name is deployADO-20230924-202539-7dae
43 Successfully deployed the template.
44 Finishing: AzureResourceManagerTemplateDeployment
```

Deploy Bicep using Infra-as-Code

www.wpninjas.eu
#WPNinjaS



Bicep tip: Optional Parameter


```
//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
  tags: {…
  }
  properties: {
    addressSpace: {
      addressPrefixes: […
      ]
    }
    dhcpOptions: {
      //using the logical operator: conditional expression (condition ? true-value : false-value)
      //source: https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/operators-logical#conditional-expression--
      dnsServers: (empty(dnsServers) ? json('null') : dnsServers)
    }
    subnets: […
    ]
  }
}

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

Bicep tip: getSecret function


```bicep
resource kv 'Microsoft.KeyVault/vaults@2022-07-01' existing = {
  name: keyvaultName
  scope: resourceGroup(keyvaultSubscription, keyvaultResourceGroup)
}

module container 'secretsmodule.bicep' = {
  scope: rg
  name: 'sa'
  params: {
    containerName: containerName
    secretMetadata: kv.getSecret(secretName, secretVersion)
    storageAccountName: sa.name
  }
}
```

We love Feedback

https://workplaceninjasummit2023.sched.com/

Workplace Ninja Summit 2023

Thank you