



Using Bicep for your Azure Virtual Desktop deployments

Starter Kit



*Workplace Ninja
Summit 2023*



Diamond Sponsor



Platinum Sponsors



Gold Sponsors



Silver Sponsors





About Esther Barthel

www.wpninjas.eu
#WPNinjaS

Focus

Cloud DevOps Engineer



From

The Netherlands



My Blog

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



Certifications



Hobbies



Contact

@virtuEs_IT

github.com/cloud-devops-ninja



About Freek Berson

www.wpninjas.eu
#WPNinjaS

Focus

End User Computing, Infrastructure as Code



From

The Netherlands

My Blog

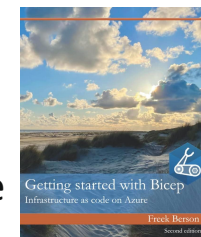
Themicrosoftplatform.net

Certifications



Books

- Getting started with Bicep
- RDS - The Complete Guide



Contact

@fberson

github.com/fberson

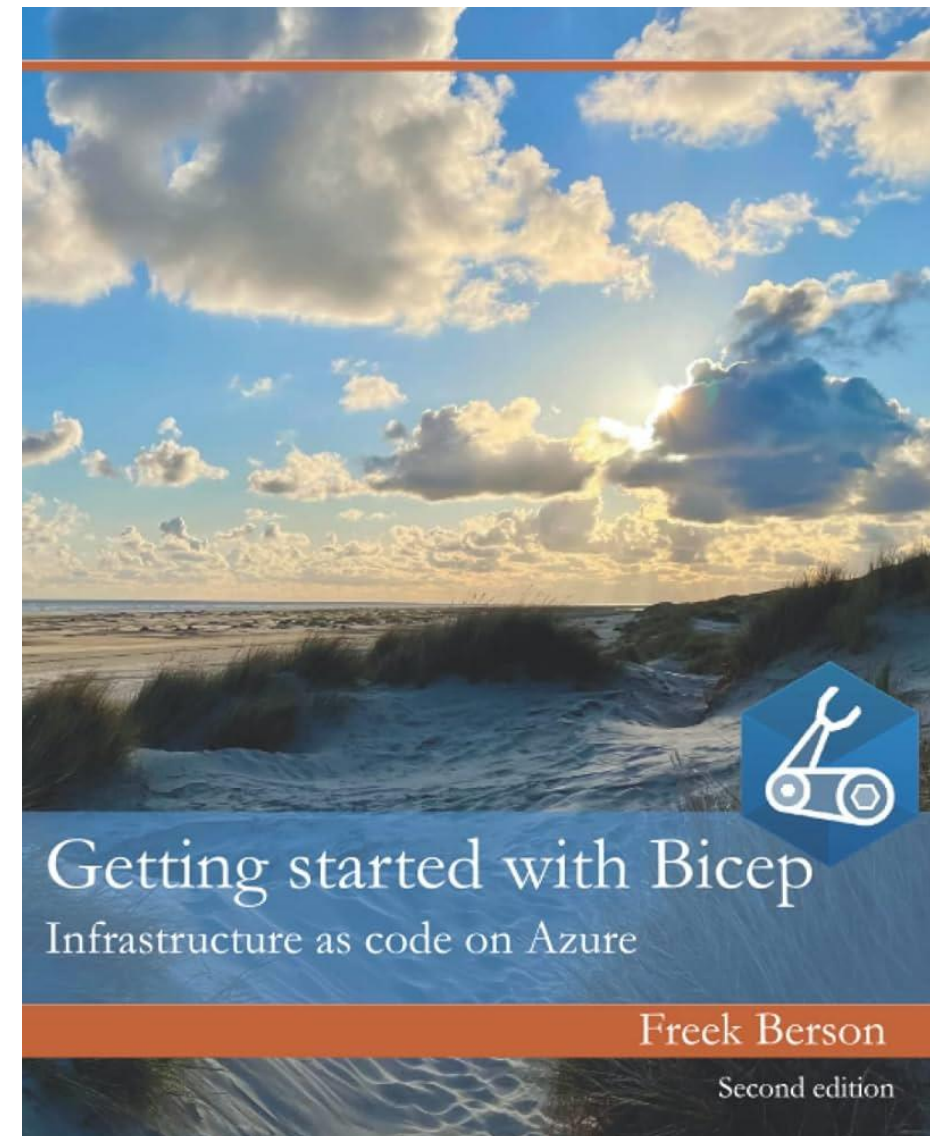


Tweet to win!

www.wpninjas.eu
#WPNinjaS

Tweet about this session and win the book!

@fberson
@virtuEs_IT
#WPNinjaS





Agenda

www.wpninjas.eu

#WPNinjaS



Key takeaways:

- Learn how to get started with Bicep
- Get the most out of VSCode integrations
- Authoring your first AVD Bicep template



Quick Introduction to IaC for Azure

Infrastructure-as-Code definition



Quick Introduction to Bicep

Bicep introduction



Demo 1: Bicep from Scratch

Creating your first Bicep file



Demo 2: Optimizing authoring experience

Decorators, linter rules, scaffolding, code snippets

bicepparam, decompile, reverseengineer



Demo 3: Expanding your Bicep skills

Conditions, Loops, AVD examples, modules, deployment pane

+ Sneak peek at tomorrow's content! 😊

Infrastructure-as-Code

code is everything



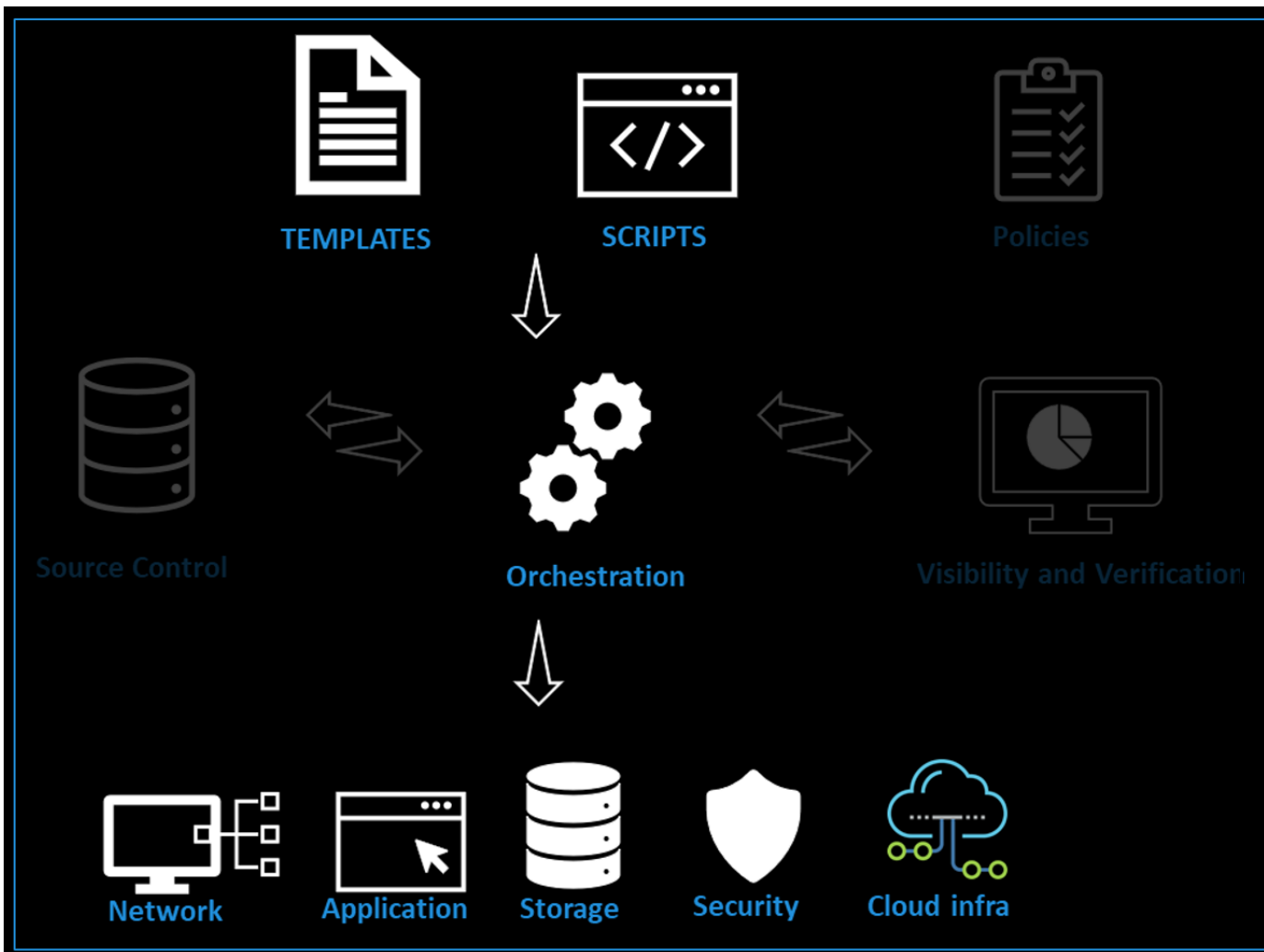


Infrastructure as Code (IaC)

www.wpninjas.eu

#WPNinjaS

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



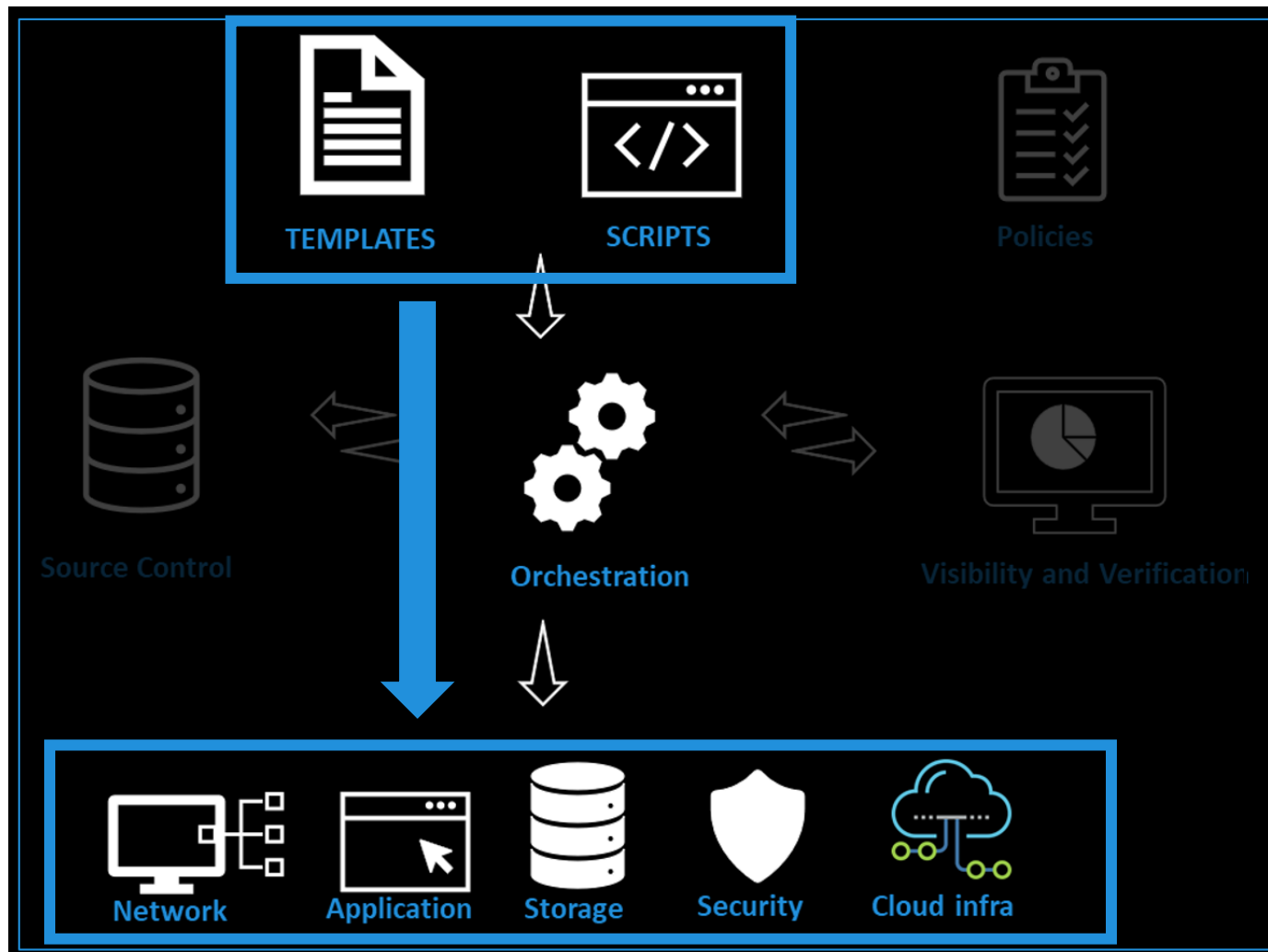


Infrastructure as Code (IaC)

www.wpninjas.eu

#WPNinjaS

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





Imperative vs Declarative code

scripts vs templates



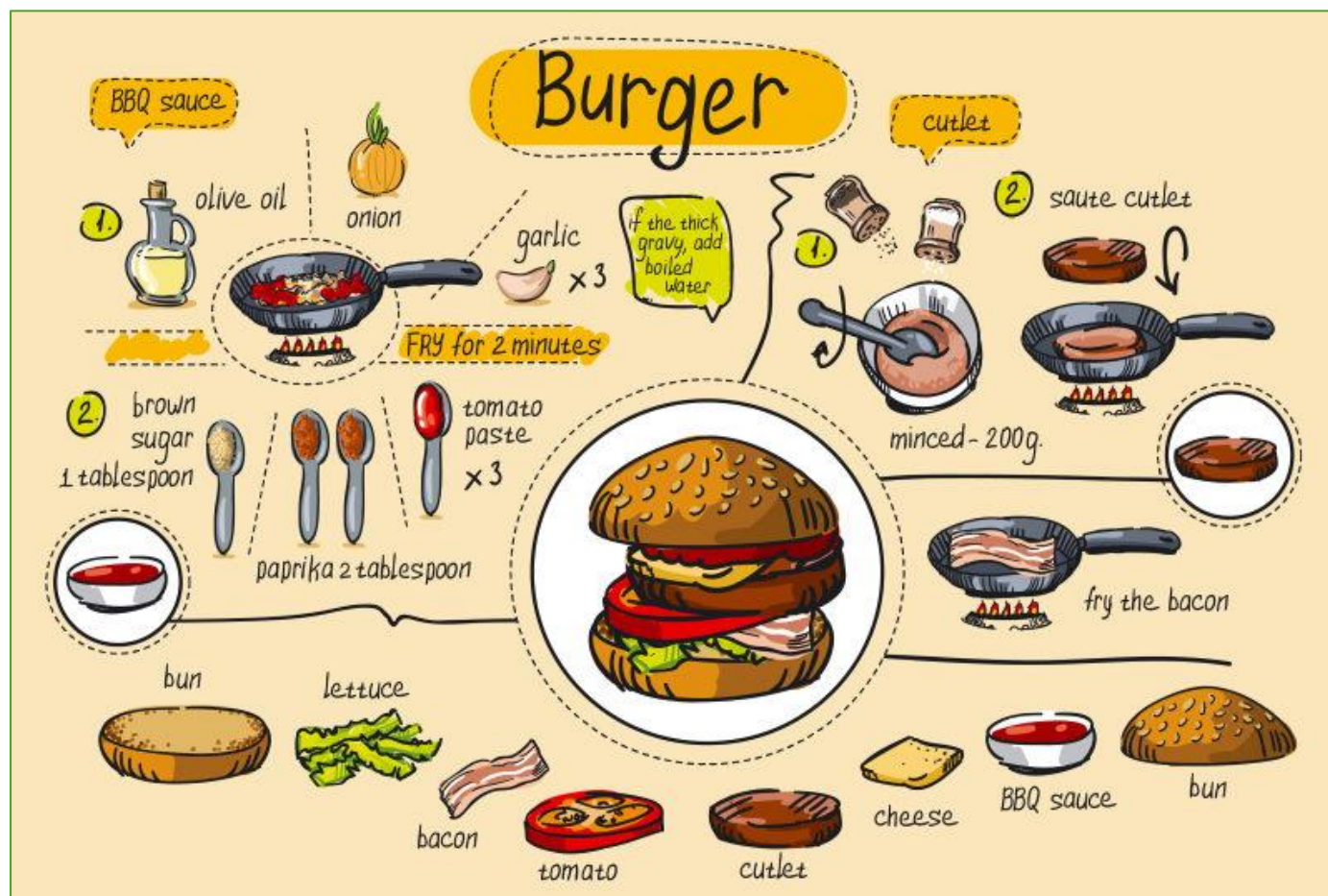
Imperative code

www.wpninjas.eu

#WPNinjaS

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**.





Imperative code

www.wpninjas.eu

#WPNinjaS

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-iac-lab-example" `
                    -Location "westeurope"
```

```
# Step 2: Create a new storage account in the resource group
```

```
New-AzStorageAccount -Name "saiac-lab01" `
                    -ResourceGroupName "rg-iac-lab-example" `
                    -SkuName Standard_LRS `
                    -Kind StorageV2 `
                    -AccessTier Cool `
                    -EnableHttpTrafficOnly
```





Declarative code

www.wpninjas.eu

#WPNinjaS

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

Like ordering from a menu.

**COWBOY RANCH
TAVERN DOUBLE™**



**HAYSTACK
TAVERN DOUBLE™**



**SIMPLY GRILLED
CHICKEN**



**BACON
CHEESEBURGER**



**SMOKE &
PEPPER™**



**WHISKEY
RIVER® BBQ**





Declarative code

www.wpninjas.eu

#WPNinjaS

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: 'saiac1ab01'  
  location: 'westeurope'  
  sku: {  
    name: 'Standard_LRS'  
  }  
  kind: 'StorageV2'  
  properties: {  
    accessTier: 'Cool'  
    supportsHttpsTrafficOnly: true  
  }  
  tags: {  
    environment: 'iac-lab'  
  }  
}
```






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

www.wpninjas.eu

#WPNinjaS

```
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",
```





introducing 'Bicep'

putting the fun back in ARM template creation



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."







Project 'Bicep'

www.wpninjas.eu
#WPNinjaS



 **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"



infoworld.com
Microsoft flexes Bicep to strengthen ARM
Azure gets a new infrastructure as code language that can help deploy and manage complex architectures

10:23 PM · Sep 8, 2020 · TweetDeck

162 Retweets 21 Quote Tweets 458 Likes



Where to position Bicep?

www.wpninjas.eu

#WPNinjaS

Intuitive

Easy to read and to author

Transpiles to ARM Templates

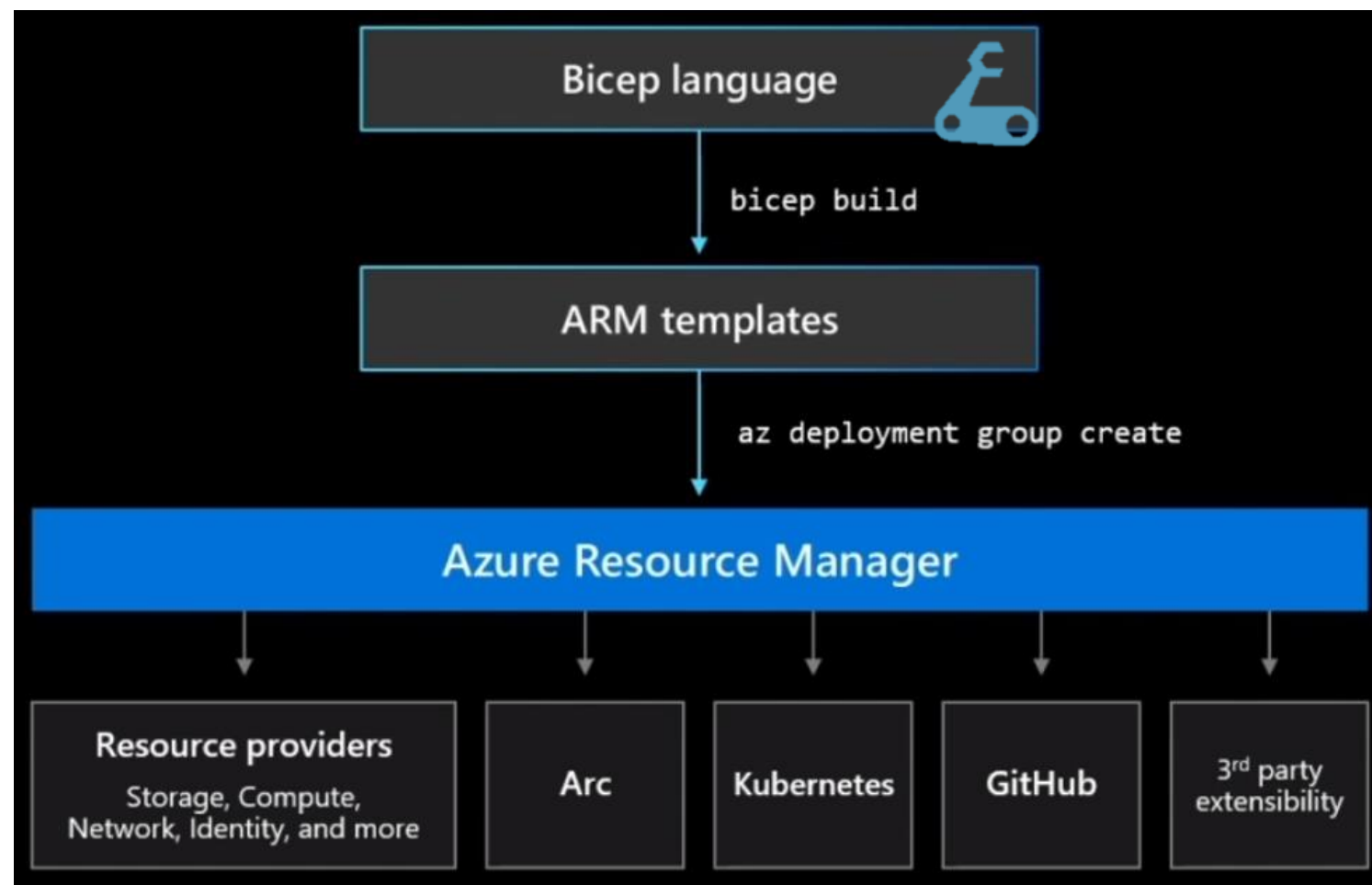
Leverage ARM template knowledge and investments

Modular

Abstract common blocks of code into reusable parts

Open Source

Transparency and community



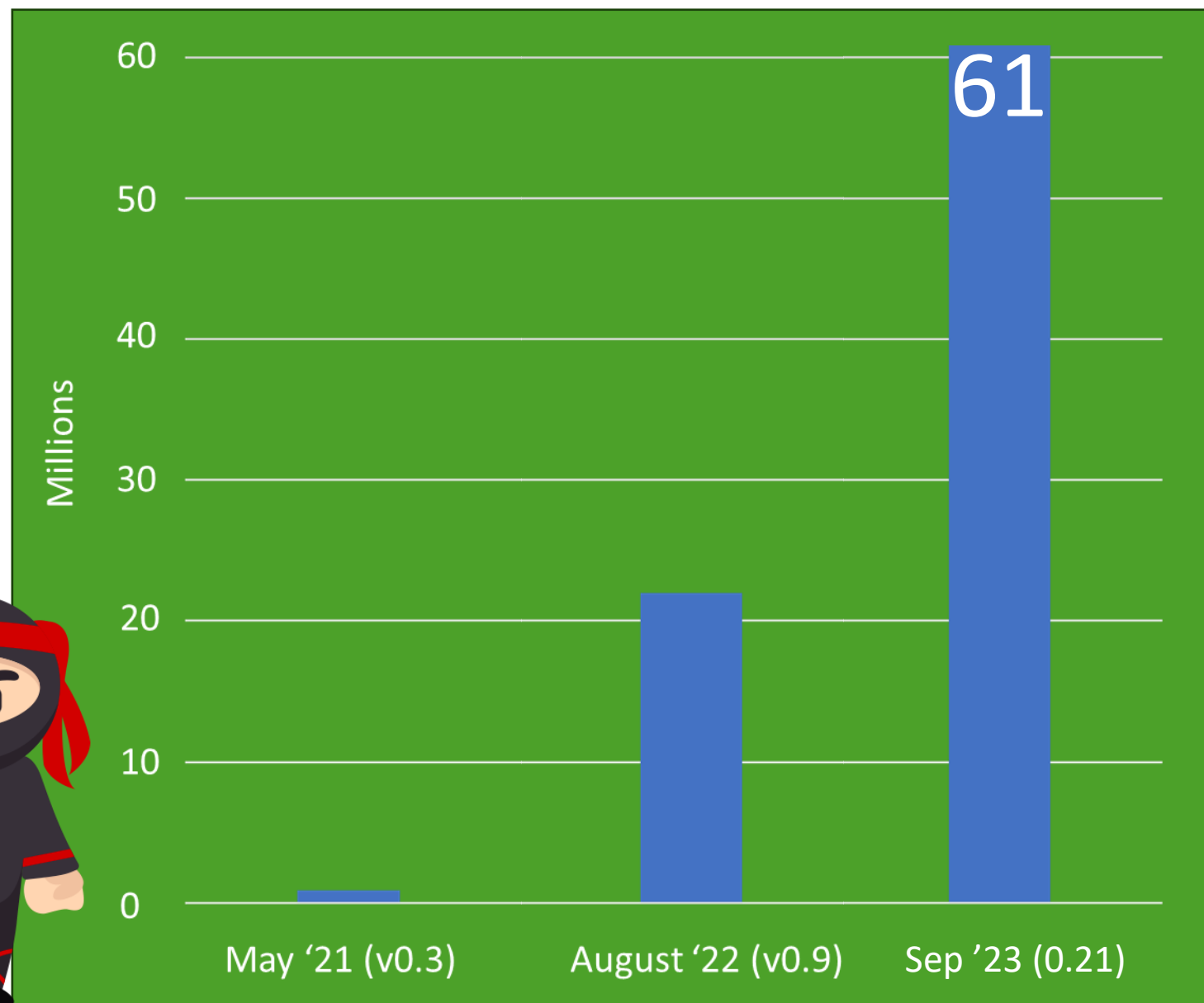


How well is Bicep adopted?

www.wpninjas.eu

#WPNinjaS

Number of resources
deployed using Bicep
in last 30 days



Getting started with Bicep

<https://aka.ms/Bicep>





Getting started with Bicep!

www.wpninjas.eu

#WPNinjaS

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

<https://aka.ms/bicep>



+



Visual Studio Code



Getting started with Bicep!

www.wpninjas.eu

#WPNinjaS

2. Create a Bicep template

The screenshot shows the Visual Studio Code interface. On the left, the 'EXTENSIONS: MARKETPLACE' sidebar is open, displaying the 'Bicep' extension by Microsoft. The search bar contains 'bicep'. The main editor area shows a file named '1. NewFile.bicep' with the following content:

```
1 param hostpoolName string = 'bicep-avd-hostpool'
2 @allowed(['BYODesktop', 'Pooled', 'Personal'])
3 param hostpoolType string = 'Pooled'
4 @allowed(['BreadthFirst', 'DepthFirst', 'Persistant'])
5 param loadBalancerType string = 'BreadthFirst'
6 @allowed(['Desktop', 'None', 'RailApplications'])
7 param preferredAppGroupType string = 'Desktop'
8
9
10 resource hp 'Microsoft.DesktopVirtualization/hostPools@2023-07-07-preview' = {
11   name: hostpoolName
12   properties: {
13     hostPoolType: hostpoolType
14     loadBalancerType: loadBalancerType
15     preferredAppGroupType: preferredAppGroupType
16   }
17 }
```

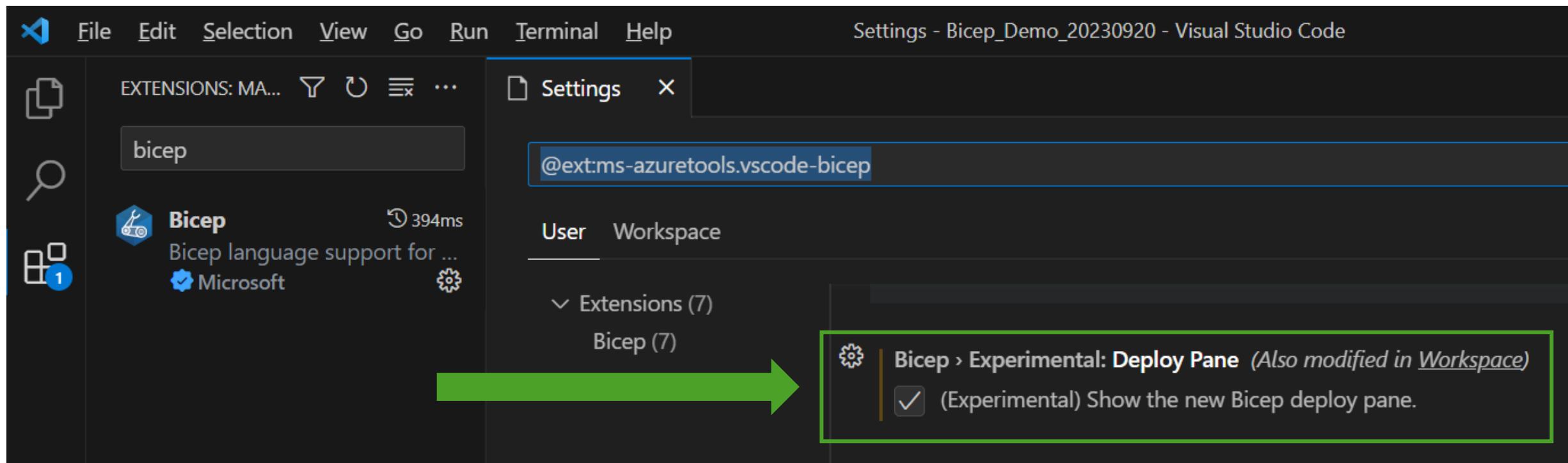


Getting started with Bicep!

www.wpninjas.eu

#WPNinjaS

3. Deploy the resources to Azure (using VSCode)





Getting started with Bicep!

www.wpninjas.eu

#WPNinjaS

3. Deploy the resources to Azure (using VSCode)

The screenshot shows the Visual Studio Code interface with a Bicep file named `1. NewFile.bicep` open. The file contains the following code:

```
1 param hostpoolName string = 'bicep-avd-hostpool'
2 @allowed(['BYODesktop', 'Pooled', 'Personal'])
3 param hostpoolType string = 'Pooled'
4 @allowed(['BreadthFirst', 'DepthFirst', 'Persistant'])
5 param loadBalancerType string = 'BreadthFirst'
6 @allowed(['Desktop', 'None', 'RailApplications'])
7 param preferredAppGroupType string = 'Desktop'
8
9
10 resource hp 'Microsoft.DesktopVirtualization/hostPools@2023-07-07-preview' = {
11   name: hostpoolName
12   properties: {
13     hostPoolType: hostpoolType
14     loadBalancerType: loadBalancerType
15     preferredAppGroupType: preferredAppGroupType
16   }
17 }
18
19
```

The deployment interface on the right side of the screen shows the following sections:

- Deployment Scope:** Includes a `Pick Scope` button and input fields for `Subscription Id` and `Resource Group` (currently set to `rg-avd-demo-wpns`).
- Parameters:** Includes a `Pick Parameters File` button and input fields for `hostpoolName` (set to `bicep-avd-hostpool`), `hostpoolType` (set to `Pooled`), `loadBalancerType` (set to `BreadthFirst`), and `preferredAppGroupType` (set to `Desktop`).
- Actions:** Includes buttons for `Deploy`, `Validate`, and `What-If`.

A green arrow points from the bottom of the code editor towards the `Deploy` button.



Demo time!

www.wpninjas.eu
#WPNinjaS



1. Creating a Bicep template from scratch
2. Optimize authoring experience with VSCode
3. Building AVD Bicep templates



Demo: scaffolding & snippets

www.wpninjas.eu

#WPNinjaS

1. scaffolding-snippets.bicep ×



Thursday > 1. scaffolding-snippets.bicep

```
1 // demo: scaffolding (AVD hostpool)
2 |
3
4 // demo: snippets (AVD hostpool)
5
```

I

Demo: parameters

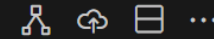
2.1. modlair-add-params.bicep ×

Thursday > 2.1. modlair-add-params.bicep > ...

```
1 // params: string
2 param hostpoolName string = 'hp-demo-wpns'
3 param hostpoolType string = 'Pooled'
4 param loadBalancerType string = 'BreadthFirst'
5 param preferredAppGroupType string = 'Desktop'
6
7 // params: boolean
8 param enableValidationMode bool = false
9
10 // params: integer
11 param maxSessionLimit int = 10
12
13 // resource: AVD hostpool
14 resource hp 'Microsoft.DesktopVirtualization/hostPools@2023-07-07-preview' = {
15   name: hostpoolName
16   properties: { param hostpoolType: string
17                 hostPoolType: hostpoolType
18                 loadBalancerType: loadBalancerType
19                 preferredAppGroupType: preferredAppGroupType
20                 validationEnvironment: enableValidationMode
21                 maxSessionLimit: maxSessionLimit
22   }
23 }
24
```

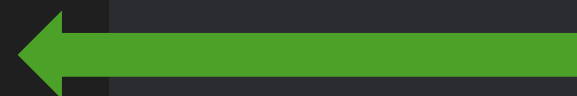
Demo: decorators

2.2. modulaire-add-decorators.bicep 2 X



Thursday > 2.2. modulaire-add-decorators.bicep > hostpoolName

```
1 param hostpoolName string = 'hp-demo-wpns'
2
3 // decorator: select (allowed values) list
4 @allowed(['BYODesktop', 'Pooled', 'Personal'])
5 param hostpoolType string = 'Pooled'
6 @allowed(['BreadthFirst', 'DepthFirst', 'Persistant'])
7 param loadBalancerType string = 'BreadthFirst'
8 @allowed(['Desktop', 'None', 'RailApplications'])
9 param preferredAppGroupType string = 'Desktop'
10
11 param enableValidationMode bool = false
12
13 // decorator: value (min-max) range
14 @minValue(1)
15 @maxValue(100)
16 param maxSessionLimit int = 10
17
18 // decorator: secure string (and linter rules)
19 @secure()
20 param secretPassword string = 'supersecret'
21
22 // resource: AVD hostpool
23 resource hp 'Microsoft.DesktopVirtualization/hostPools@2023-07-07-preview' = {
24   name: hostpoolName
25   properties: {
26     hostPoolType: hostpoolType
27     loadBalancerType: loadBalancerType
```





Demo: decorators

2.2. modlair-add-decorators.bicep 2 x



Thursday > 2.2. modlair-add-decorators.bicep > ...

```
13 // decorator: value (min-max) range
14 @minValue(1)
15 @maxValue(100)
16 param maxSessionLimit int = 10
17
18 // decorator: secure string (and linter rules)
19 @secure()
20 param secretPassword string = 'supersecret'
21
22 // resource: AVD hostpool
23 resource hp 'Microsoft.DesktopVirtualization/hostPools@2023-07-07-preview' = {
24   name: hostpoolName
25   properties: {
26     hostPoolType: hostpoolType
27     loadBalancerType: @minValue(1)
28     preferredAppGroup: @maxValue(100)
29     validationEnvironment: param maxSessionLimit: int
30     maxSessionLimit: maxSessionLimit
31   }
32 }
```



Demo: linter rules

2.2. modlair-add-decorators.bicep 2 ✕

Thursday > 2.2. modlair-add-decorators.bicep

```
15 @maxVa
16 param
17 @secure()
18 // dec
19 @secur
20 param secretPassword string = 'supersecret'
21
```

Parameter "secretPassword" is declared but never used. bicep
core(<https://aka.ms/bicep/linter/no-unused-params>)
View Problem (Alt+F8) Quick Fix... (Ctrl+.)

2.2. modlair-add-decorators.bicep 2 ✕

Thursday > 2.2. modlair-add-decorators.bicep > ...

```
15 @maxValue(100)
16 param maxSessionLimit
18 // decorator: secure
19 @secure()
20 param secretPassword string = 'supersecret'
21
```

Secure parameters should not have hardcoded defaults (except for empty or newGuid()). bicep
core(<https://aka.ms/bicep/linter/secure-parameter-default>)
View Problem (Alt+F8) Quick Fix... (Ctrl+.)



Demo time!

www.wpninjas.eu
#WPNinjaS

- ✓ 1. Creating a Bicep template from scratch
- ✓ 2. Optimize authoring experience with VSCode
- 3. Building AVD Bicep templates





Hope to see you tomorrow!

www.wpninjas.eu

#WPNinjaS

Friday
14:00



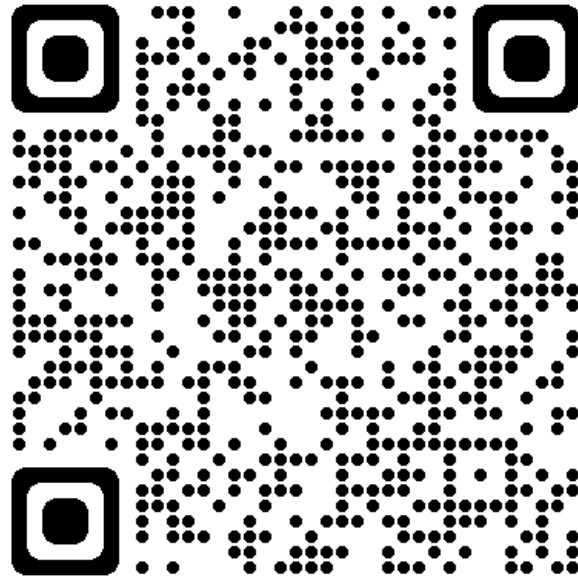
Using Bicep for your Azure Virtual Desktop deployments - Masterclass



Visualizer, Modules, Module Registries, deployment Stacks,
Graph API ^{Preview}



Deploy Bicep to Azure: PowerShell, CLI, and Azure DevOps,
working with secure values



We love Feedback

<https://workplaceninjasummit2023.sched.com/>





Thank you

