

Don't be left behind with the ATA Learning Newsletter!

Your email address

Subscribe Today!



Getting Started with Azure Bicep (Step-by-Step)

Published: 29 October 2021 – 5 min. read

MICROSOFT AZURE



Nick Rimmer

Read [more tutorials](#) by Nick Rimmer!



veeAM

Ransomware recovery test drive: This technical workshop is designed to take you behind the scenes and shows you how to adopt recovery, ensuring you're ready. REQUEST YOUR LAB

Your Job!

Don't be left behind with the ATA Learning Newsletter!



Prerequisites

What is Azure Bicep?

Setting Up a Bicep Development Environment

- Installing the VS Code Bicep Extension
- Installing the Bicep CLI

Creating a Bicep Template to Deploy a Storage Account

Deploying a Bicep Template with PowerShell

Deploying a Bicep Template with the Azure CLI

- Conclusion

Do you need to build infrastructure in Azure? Have you looked at ARM templates and thought, whoa, looks complex? Because many people complained about the difficulty of building ARM templates, Microsoft developed a layer of abstraction called Azure Bicep.

In this tutorial, you will learn how to get started from scratch. You'll learn how to do everything from setting up a development environment to deploying your first Bicep template!



Let's get started!

Don't be left behind with the ATA Learning Newsletter!



Prerequisites

This tutorial will be a hands-on demonstration. If you'd like to follow along, be sure you have the following:

- Windows 10 – The majority of the demos in this tutorial will work on other operating systems but all demos will use Windows 10.
- Visual Studio Code – This tutorial will use v1.60.2.

Related: [What You Need to Know about Visual Studio Code: A Tutorial](#)

- An Azure Resource Group to deploy to – This tutorial will use a resource group called ATA.

What is Azure Bicep?

Bicep is a domain-specific language that makes building ARM templates easier. You can think of Bicep as an abstraction layer on top of ARM templates. ARM uses JSON and can get complex, especially for large deploys.

Don't be left behind with the ATA Learning Newsletter!



Microsoft decided to create Bicep to reduce the complexities and improve the development experience. It is not a new language; think of it more as an improvement to the JSON ARM templates.

Setting Up a Bicep Development Environment

To start developing Bicep templates, you need to set up an environment that will help you get the best results. To do that, you have a few tools at your disposal. Let's first set up the environment you need including the VS Code Bicep extension, the Bicep CLI, and optionally the Azure PowerShell modules.

Installing the VS Code Bicep Extension

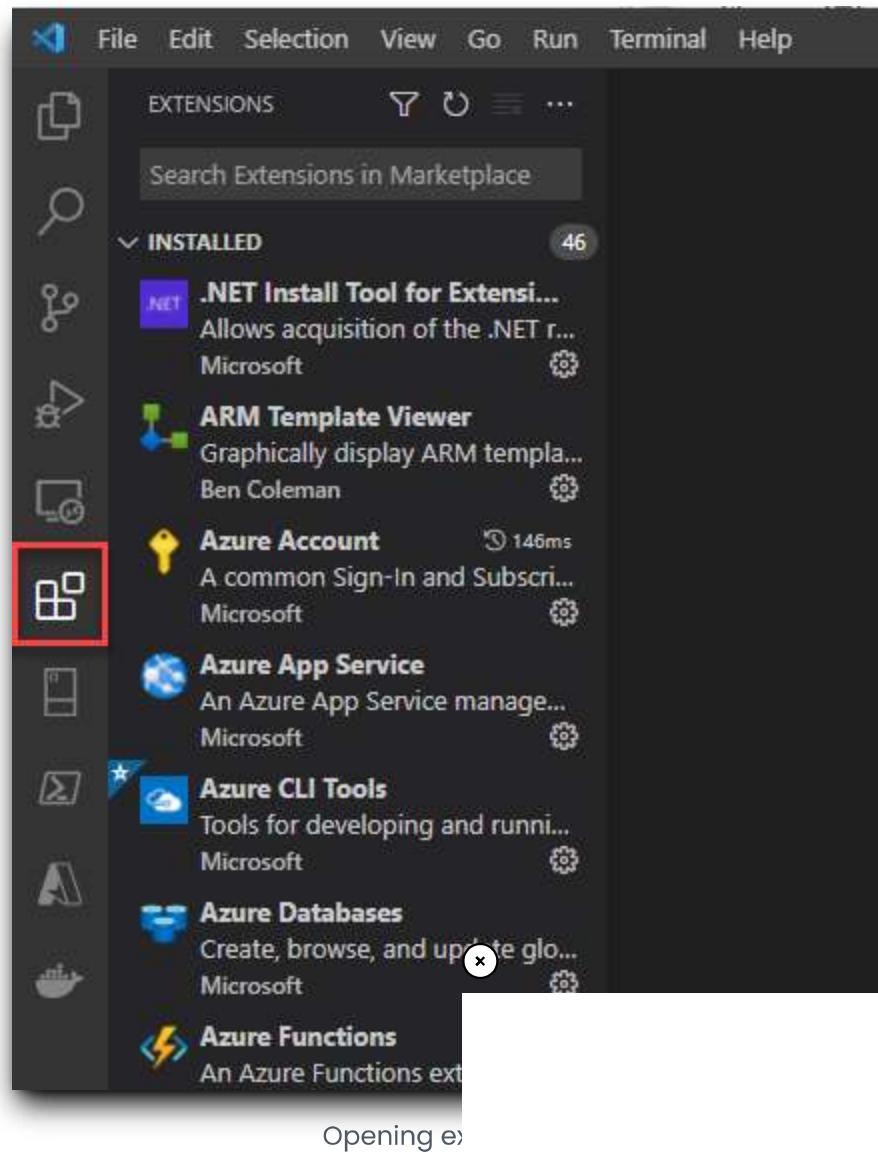
VS Code is a great development tool to build Bicep templates. As such, Microsoft provides a VS Code extension to help make creating Bicep templates easier. Let's start building a development environment by prepping VS Code.



Don't be left behind with the ATA Learning Newsletter!



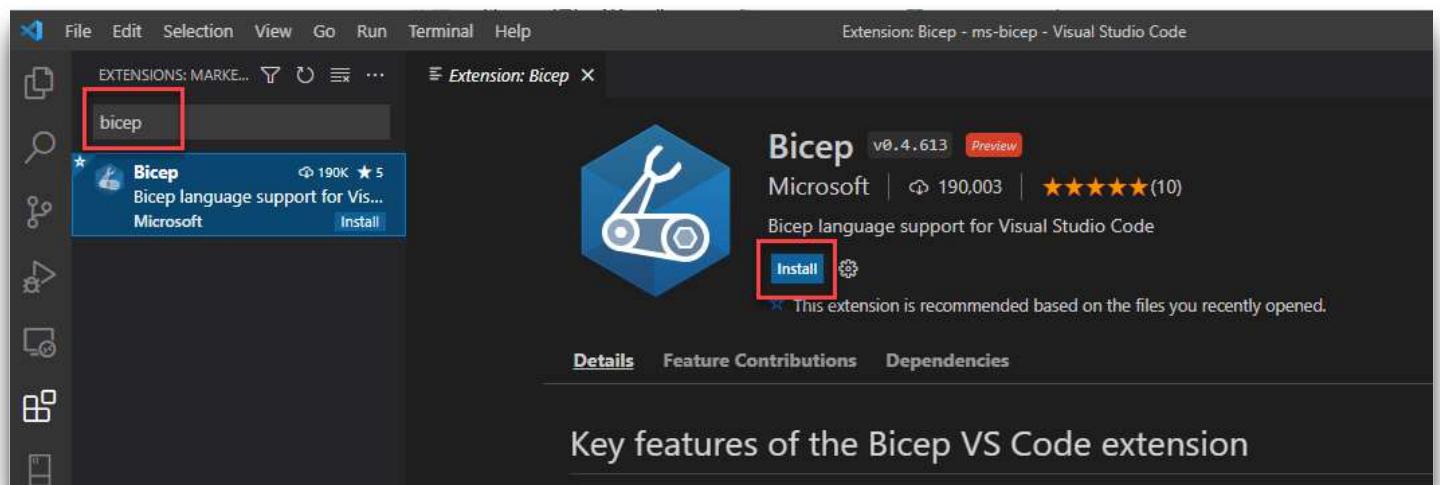
Open Visual Studio Code and click on the Extensions button.



Don't be left behind with the ATA Learning Newsletter!



In the search bar, type “bicep”. You should see a single **Bicep** extension appear. When this happens, click on **Install**. After a few moments, VS Code will install the Bicep extension.



Search and install bicep extension

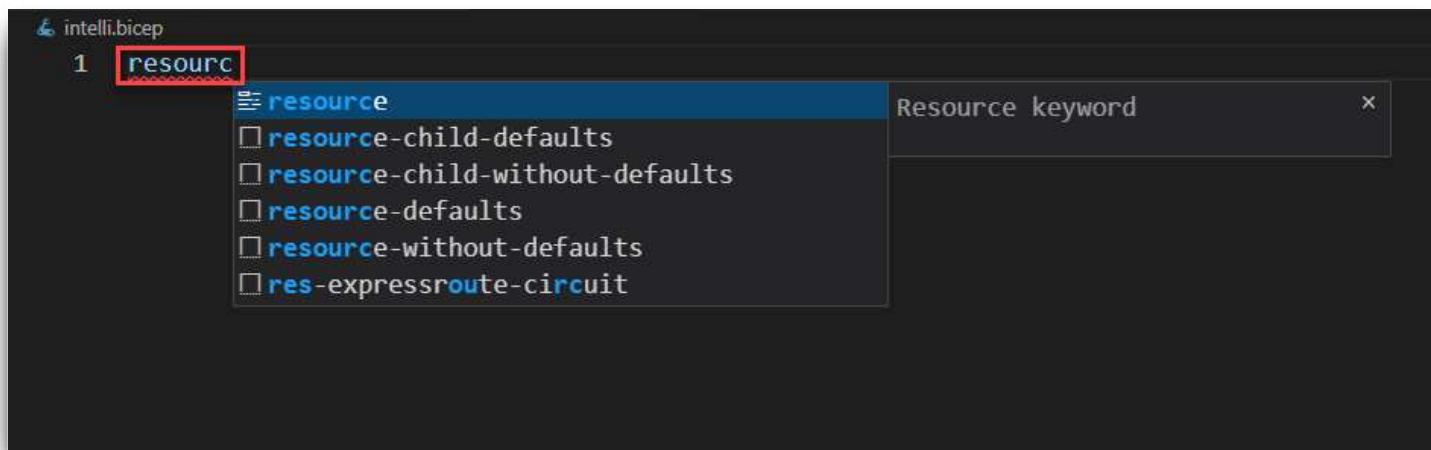


Don't be left behind with the ATA Learning Newsletter!



undoubtedly have times when you need to create Bicep templates from scratch. The Bicep extension in VS Code comes in handy here particularly due to its Intellisense features.

Notice that if you open a file with an extension of `.bicep` in VS Code, VS Code will attempt to provide shortcuts for you. For example, you create Azure resources with the `resource` keyword in a Bicep template. With the Bicep extension installed, VS Code understands this.



Typing the resource command

Don't be left behind with the ATA Learning Newsletter!



```
↳ intelli.bicep > {} Identifier
1   resource Identifier 'Provider/ParentType/ChildType@Version' = {
2     name: |
3     properties: {
4       |
5     }
6 }
```

Blank bicep template

The period (.) is another handy shortcut when creating Bicep templates. By typing this in the **properties** section, for example, you'll see all the properties available for a storage account.



Don't be left behind with the ATA Learning Newsletter!

The screenshot shows a code editor with a tooltip displayed over a specific property in a Bicep configuration file. The code snippet is as follows:

```
8 kind: 'StorageV2'  
9 properties: [  
10   accessTier: 'Hot'  
11   .:  
12     allowBlobPublicAccess  
13   }  
14 }  
15  
TERMINAL PROBL  
PS D:\git\ms  
PS D:\git\ms
```

The tooltip for the `allowBlobPublicAccess` property is shown, highlighting its type and description:

`allowBlobPublicAccess` x
Type: `bool`
Allow or disallow public access to all blobs or containers in the storage account. The default interpretation is true for this property.

Properties available for a storage account



Don't be left behind with the ATA Learning Newsletter!



```
az bicep version
```

If you have the Bicep CLI installed, you'll see a version statement, as shown below.

```
PS D:\git\ms-bicep> az bicep version
Bicep CLI version 0.4.13 (d826ce8411)
```

Bicep CLI

If you see an error message about the Bicep CLI command, run the `az bicep init` subcommand to install the Bicep CLI.



Don't be left behind with the ATA Learning Newsletter!



Bicep installed successfully

As an alternate installation method, you can also install the Bicep CLI via the [Bicep Windows Installer](#).



Creating a Bicep Template to

nt

You can deploy some complex Azure resources using Bicep. To demonstrate, let's create a simple Azure storage account and deploy it to Azure using both the Azure CLI and I

r your
use to

Don't be left behind with the ATA Learning Newsletter!



- The resource identifier (`resource myStorage`) – Tells Bicep to create a new resource named `myStorage`. This name identifies the particular resource within the Bicep template. This name is not the name of the resource that is created in Azure.
- `Microsoft.Storage/storageAccounts@2019-06-01` – Defines a type composed of the resource provider `Microsoft.Storage`, the resource type (`storageAccounts`), and the API version (`2019-06-01`) to use.
- `name` – The name of the resource as it appears in Azure (pay attention to the rules and restrictions for naming Azure resources.)
- `location` – The Azure region to create the resource in.



Don't be left behind with the ATA Learning Newsletter!



```
resource myStorage 'Microsoft.Storage/storageAccounts@2019-06-01' = {
    name: 'ata2021bicepdiskstorage'
    location: 'uksouth'
    SKU: {
        name: 'Standard_LRS'
    }
    kind: 'StorageV2'
    properties: {
        accessTier: 'Hot'
    }
}
```

Deploying a Bicep Template with PowerShell

As mentioned earlier, you can deploy Bicep templates one of two ways; via the Azure CLI and Azure PowerShell module. Let's first deploy a Bicep template with PowerShell.



Don't be left behind with the ATA Learning Newsletter!



Deploying Azure resources with Bicep templates is identical to deploying ARM templates. Microsoft has essentially added support for the Bicep language to many of the existing PowerShell ARM cmdlets such as `New-AzResourceGroupDeployment`.

In your PowerShell console, invoke a new resource group deployment via the `New-AzResourceGroupDeployment` cmdlet providing the path of the Bicep template and the `ResourceGroupName` to deploy to. You'll notice that you can use a Bicep template exactly how you could an ARM template.

```
New-AzResourceGroupDeployment -TemplateFile main.bicep -ResourceGroupName ATA
```

If the deployment is successful, you should see the following output in your PowerShell session with the property `ProvisioningState` showing as successful. That's it!

```
PS D:\git\ms-bicep> New-AzResourceGroupDeployment -TemplateFile main.bicep -ResourceGroupName ATA

DeploymentName      : main
ResourceGroupName   : ATA
ProvisioningState   : Succeeded
Timestamp          : 15/10/2021 17:45:06
Mode               : Incremental
TemplateLink       :
Parameters         :
Outputs            :
DeploymentLogLevel :
```

Deploying resou

Don't be left behind with the ATA Learning Newsletter!



To verify the resource was created successfully, log into the Azure Portal, navigate to your resource group, and you will see inside that resource group a newly created storage account.

The screenshot shows the Azure Portal's 'Resource groups' view for the 'ATA' resource group. The 'Essentials' section provides an overview of the subscription (Move: Pay-As-You-Go, ID: 19b3488b-1b19-4127-8f27-8cfb2b594615), deployment status (1 Succeeded), and location (UK South). Below this, the 'Resources' section lists a single item: a Storage account named 'ata2021bicepdiskstorage' located in 'UK South'. The entire screenshot is framed by a large red border.

If you're following along and created the storage account, remove it! Notice the `ResourceName` coincides with the template.



Don't be left behind with the ATA Learning Newsletter!



```
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PowerShell Integrated Console + × └─ PS D:\git\ms-bicep> Remove-AzResource -ResourceName ata2021bicepdiskstorage -ResourceGroupName ATA -ResourceType "Microsoft.Storage/storageAccounts"
Confirm
Are you sure you want to delete the following resource: /subscriptions/2021bicepdiskstorage/resourceGroups/ATA/providers/Microsoft.Storage/storageAccounts/ata2021bicepdiskstorage
[Y] Yes [N] No [S] Suspend [?] Help (default is "Yes"): Yes
True
PS D:\git\ms-bicep>
```

Removing an Azure storage account

Deploying a Bicep Template with the Azure CLI



To round out this tutorial, let's now cover deploying our Bicep template with the Azure CLI.

Be sure you have [Azure CLI v2.2.0+](#) installed before moving on to the next section.

Don't be left behind with the ATA Learning Newsletter!



If you see the **provisioningState** as **Succeeded**, as shown below, ARM has read the Bicep template and carried out all of the instructions within!

```
PS D:\git\ms-bicep> az deployment group create --resource-group ATA --template-file main.bicep
A new Bicep release is available: v0.4.1008. Upgrade now by running "az bicep upgrade".
{
  "id": "/subscriptions//resourceGroups/ATA/providers/Microsoft.Resources/deployments/main",
  "location": null,
  "name": "main",
  "properties": {
    "correlationId": "6f9cd761-62bc-45a2-9a3b-c53f5129302a",
    "debugSetting": null,
    "dependencies": [],
    "duration": "PT20.9999235S",
    "error": null,
    "mode": "Incremental",
    "onErrorDeployment": null,
    "outputResources": [
      {
        "id": "/subscriptions//resourceGroups/ATA/providers/Microsoft.Storage/storageAccounts/atastorage",
        "resourceGroup": "ATA"
      }
    ]
  }
}
```

provisioningState

Don't be left behind with the ATA Learning Newsletter!



Output in JSON from az deployment command

Don't forget to clean up the resource with `az resource delete -g ATA -n ata2021bicepdiskstorage --resource-type "Microsoft.Storage/storageAccounts"`

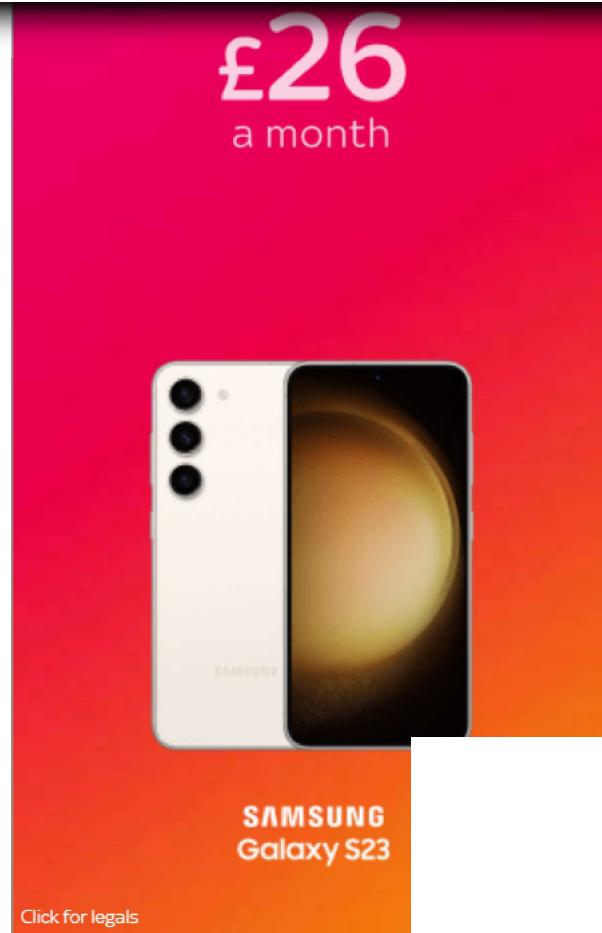
Conclusion



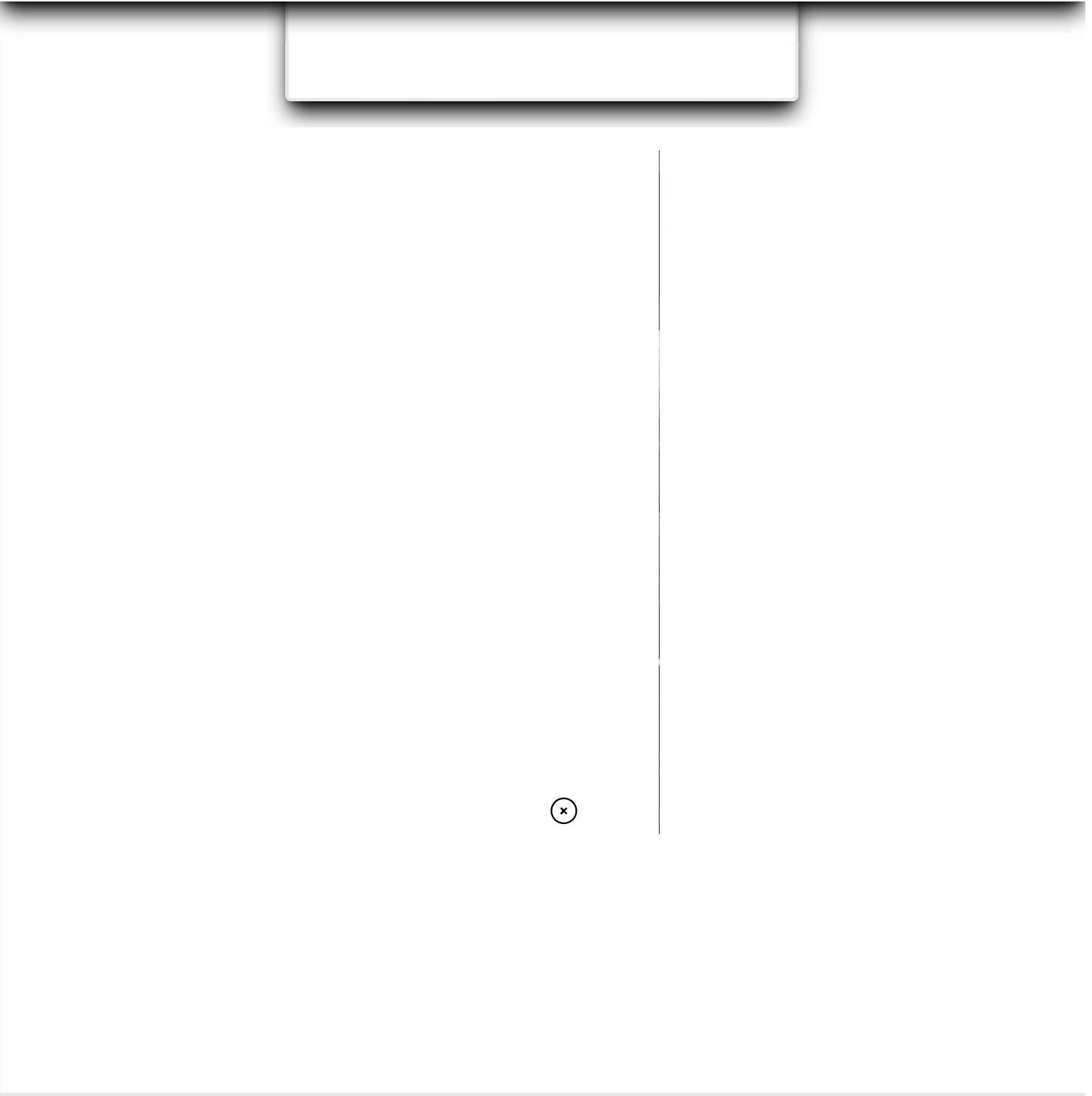
Now you have created a development environment resource in Azure, what is your next step? Perhaps offer on Bicep?

to

Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!

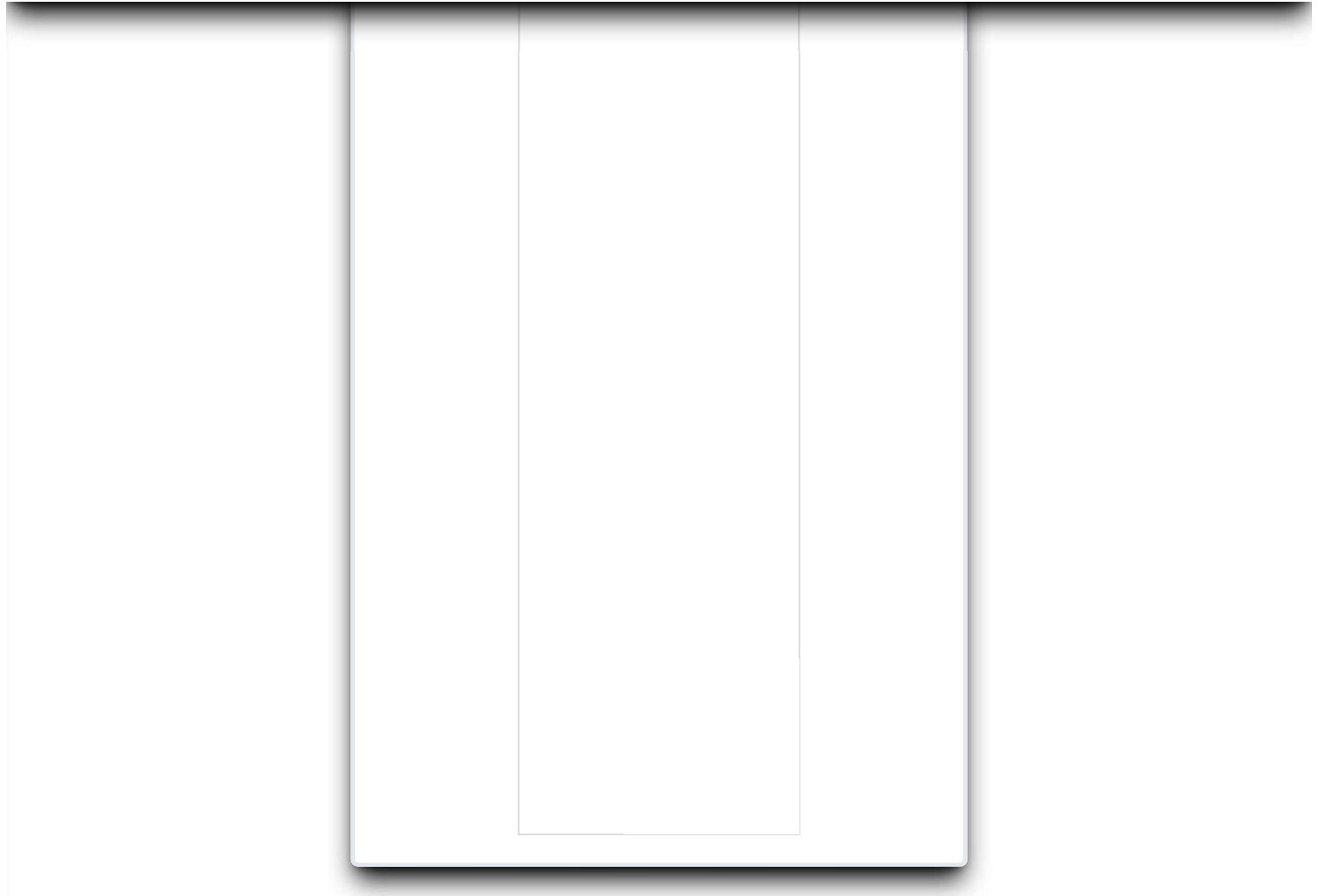


Hate ads? Want to support the writer? Get many of our tutorials packaged as an ATA Guidebook.

Explore ATA Guidebooks



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



Don't be left behind with the ATA Learning Newsletter!



More from ATA Learning & Partners



Recommended Resources!

Recommended Resources for
Training, Information Security,
Automation, and more!



Don't be left behind with the ATA Learning Newsletter!



junior admin or system architect,
you have something to share.
Why not write on a platform with
an existing audience and share
your knowledge with the world?



ATA Guidebooks

Continuously updated and always
available eBooks. Learn more on
LeanPub!

ATA Learning Guidebooks

ATA Learning is known for its
high-quality written tutorials in
the form of blog posts. Support
ATA Learning with ATA Guidebook
PDF eBooks available offline and
with nc

Continue learning with th

Don't be left behind with the ATA Learning Newsletter!



Cloud

- DevOps
- Home Ops
- Information Security
- Software Development

SITE

- Home
- Tutorials
- Guidebooks
- Instructors
- Get Paid to Write
- Advertising
- Recommended Resources
- Jobs



Copyright 2023© ATA Le

[Update Privacy](#)

A RAPTIVE PA