

# Flexing with Bicep

Modern Azure Infrastructure as Code

Garó Yeriázarian  
Software Whisperer

# Agenda

What are we going to cover?



Overview of Azure Resource Management



What is Bicep and why does it matter?



Infrastructure as Code patterns in the real world



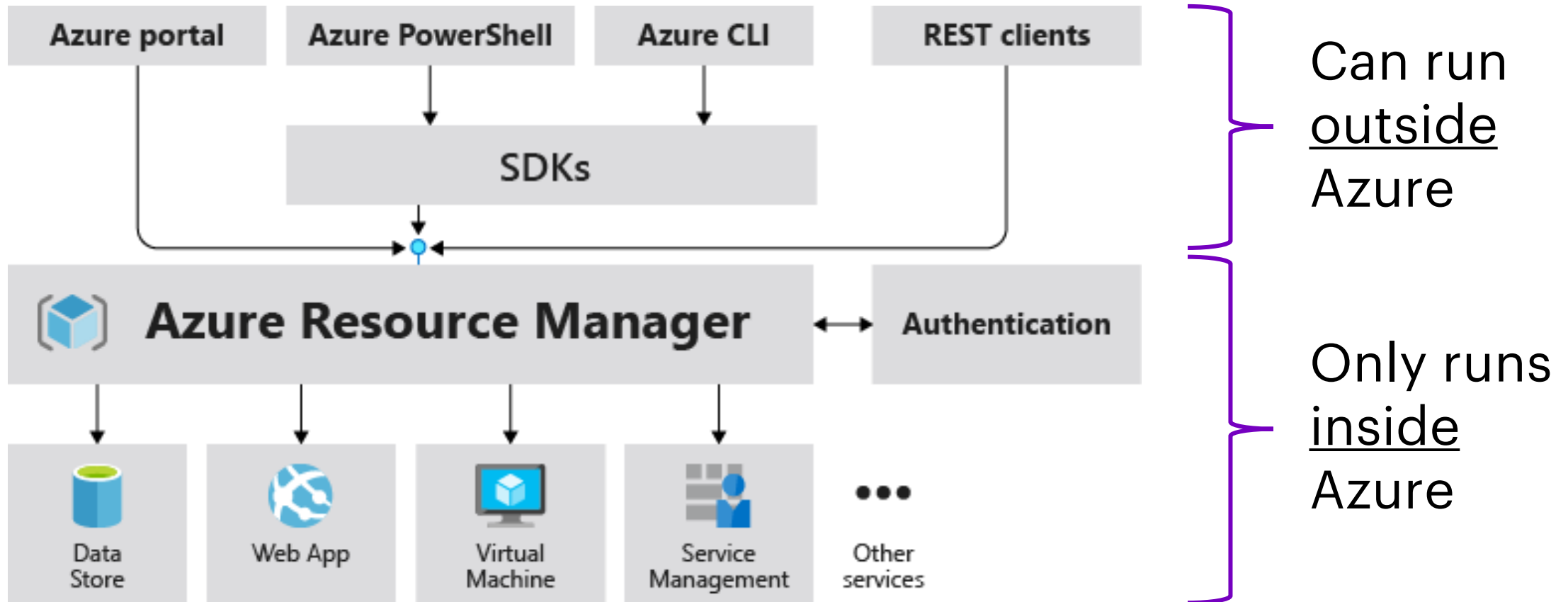
Iterative developer workflows



ASK ME QUESTIONS ANYTIME, PLEASE INTERRUPT ME!



# Azure Resource Manager (ARM)



[Azure Resource Manager overview - Azure Resource Manager | Microsoft Learn](#)

# Azure Resource Manager Templates: Why Bicep?

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2021-04-01",
      "name": "[parameters('storageAccountName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "[parameters('skuName')]"
      },
      "kind": "StorageV2",
      "properties": {
        "accessTier": "[parameters('accessTier')]"
      }
    }
  ],
  "parameters": {
    "storageAccountName": {
      "type": "string",
      "metadata": {
        "description": "Name of the storage account"
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "westus",
      "metadata": {
        "description": "Location for the storage account"
      }
    },
    "skuName": {
      "type": "string",
      "defaultValue": "Standard_LRS",
      "allowedValues": [
        "Standard_LRS",
        "Standard_GRS",
        "Standard_ZRS",
        "Premium_LRS"
      ],
      "metadata": {
        "description": "Storage account SKU"
      }
    },
    "accessTier": {
      "type": "string",
      "defaultValue": "Hot",
      "allowedValues": [
        "Hot",
        "Cool"
      ],
      "metadata": {
        "description": "Access tier for the storage account"
      }
    }
  }
}
```



```
@description('Name of the storage account')
param storageAccountName string

@description('Location for the storage account')
param location string = 'westus'

@description('Storage account SKU')
@allowed([
  'Standard_LRS'
  'Standard_GRS'
  'Standard_ZRS'
  'Premium_LRS'
])
param skuName string = 'Standard_LRS'

@description('Access tier for the storage account')
@allowed([
  'Hot'
  'Cool'
])
param accessTier string = 'Hot'

resource storageAccount 'Microsoft.Storage/storageAccounts@2021-04-01' = {
  name: storageAccountName
  location: location
  sku: {
    name: skuName
  }
  kind: 'StorageV2'
  properties: {
    accessTier: accessTier
  }
}
```

# DEMO

## Bicep Templates from Azure Portal

# Azure Portal: Initial View

The screenshot displays the Microsoft Azure Portal interface. At the top, the navigation bar includes the 'Microsoft Azure' logo, a search bar, and the user profile 'cloud\_user\_p\_e04b3031...'. The main content area shows the 'Overview' page for the resource group '1-548b3e2c-playground-sandbox'. The left sidebar contains a list of navigation options: Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings, Cost Management, Monitoring, Automation, and Help. The main content area features a 'Create' button and a 'Manage view' dropdown. Below this, there is a 'Resources' tab and a 'Recommendations' tab. The 'Resources' tab shows a table with columns for Name, Type, and Location. The table is currently empty, displaying the message 'No resources match your filters'. The bottom of the page shows a large gray cube icon and the text 'No resources match your filters'.

# Azure Portal: Create a Storage Account

This screenshot shows the first step of the 'Create a storage account' wizard in the Azure Portal. The breadcrumb trail is 'Home > 1-548b3e2c-playground-sandbox > Marketplace >'. The title is 'Create a storage account'. Under 'Project details', there is a description: 'Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.' The 'Subscription' dropdown is set to 'P4-Real Hands-On Labs' and the 'Resource group' dropdown is set to '1-548b3e2c-playground-sandbox', with a 'Create new' link below it. Under 'Instance details', the 'Storage account name' is 'myfavoritestoragehouston', the 'Region' is '(US) East US' (with a 'Deploy to an Azure Extended Zone' link), the 'Primary service' is 'Azure Blob Storage or Azure Data Lake Storage Gen 2', and the 'Performance' is 'Standard: Recommended for most scenarios (general-purpose v2 account)' (selected with a radio button). At the bottom are 'Previous', 'Next', and 'Review + create' buttons.

Microsoft Azure

Home > 1-548b3e2c-playground-sandbox > Marketplace >

## Create a storage account

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription \* P4-Real Hands-On Labs

Resource group \* 1-548b3e2c-playground-sandbox  
[Create new](#)

Instance details

Storage account name \* ① myfavoritestoragehouston

Region \* ① (US) East US  
[Deploy to an Azure Extended Zone](#)

Primary service ① Azure Blob Storage or Azure Data Lake Storage Gen 2

Performance \* ① ☒ Standard: Recommended for most scenarios (general-purpose v2 account)

Previous Next Review + create

This screenshot shows the 'Review + create' step of the 'Create a storage account' wizard. The breadcrumb trail is 'Home > 1-548b3e2c-playground-sandbox > Marketplace >'. The title is 'Create a storage account'. Navigation tabs include 'Basics', 'Advanced', 'Networking', 'Data protection', 'Encryption', 'Tags', and 'Review + create' (which is underlined). There is a link to 'View automation template'. The 'Basics' section contains a summary table of the configuration. The 'Advanced' section shows 'Enable hierarchical namespace' and 'Enable SFTP' both set to 'Disabled'. At the bottom are 'Previous', 'Next', and 'Create' buttons.

Microsoft Azure

Home > 1-548b3e2c-playground-sandbox > Marketplace >

## Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review + create

[View automation template](#)

**Basics**

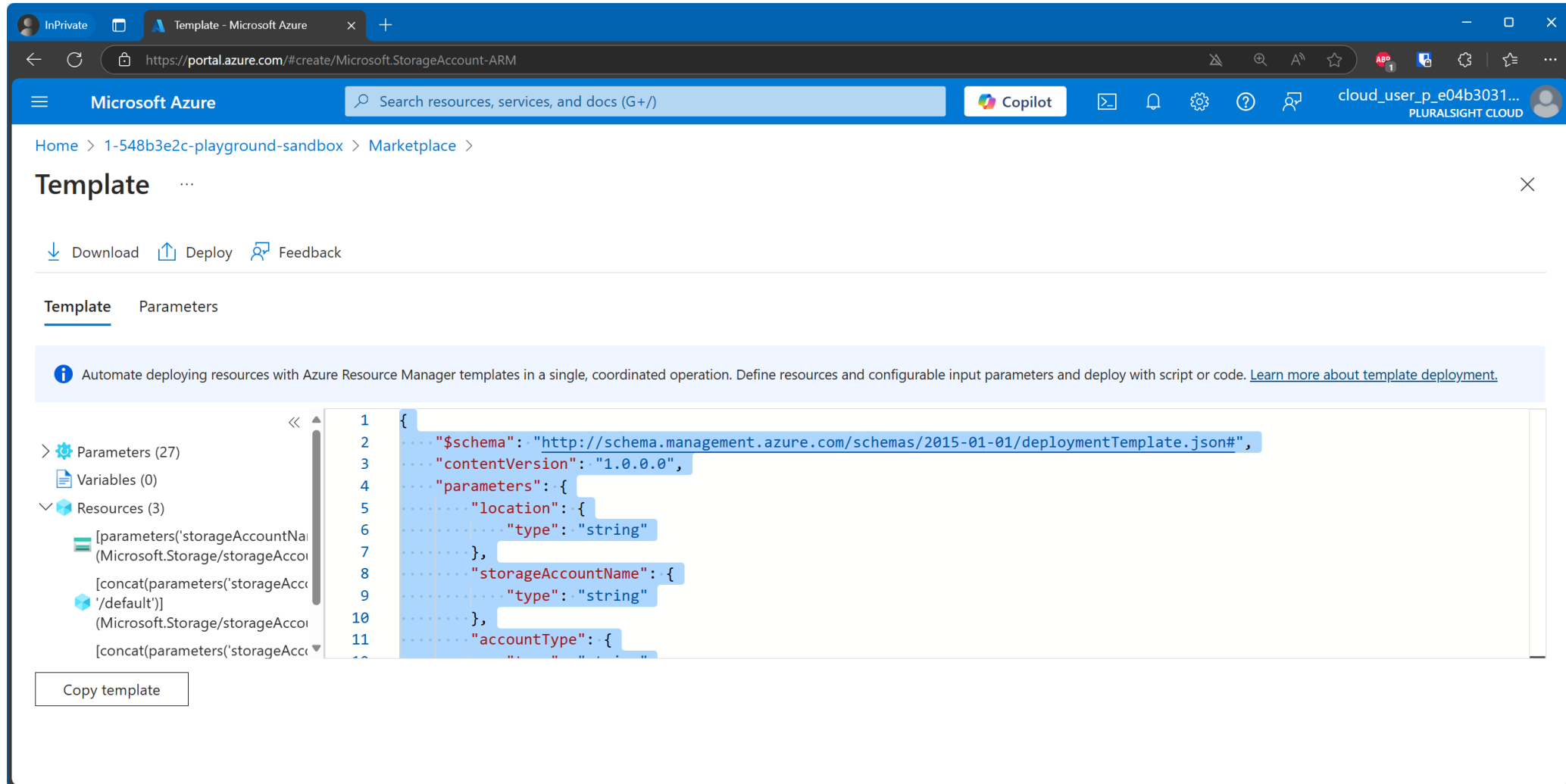
Subscription	P4-Real Hands-On Labs
Resource group	1-548b3e2c-playground-sandbox
Location	East US
Storage account name	myfavoritestoragehouston
Primary service	Azure Blob Storage or Azure Data Lake Storage Gen 2
Performance	Standard
Replication	Locally-redundant storage (LRS)

**Advanced**

Enable hierarchical namespace	Disabled
Enable SFTP	Disabled

Previous Next Create

# Azure Portal: View Automation Template



The screenshot displays the Azure Portal interface for viewing an ARM template. The browser address bar shows the URL `https://portal.azure.com/#create/Microsoft.StorageAccount-ARM`. The page title is "Template" and it includes navigation links for "Download", "Deploy", and "Feedback". The "Template" tab is active, showing a description: "Automate deploying resources with Azure Resource Manager templates in a single, coordinated operation. Define resources and configurable input parameters and deploy with script or code. [Learn more about template deployment.](#)".

On the left, a sidebar lists the template's components: "Parameters (27)", "Variables (0)", and "Resources (3)". The "Resources (3)" section is expanded, showing three resource definitions:

- `[parameters('storageAccountName')]` (Microsoft.Storage/storageAccounts)
- `[concat(parameters('storageAccountName'), 'default')]` (Microsoft.Storage/storageAccounts)
- `[concat(parameters('storageAccountName'), 'default')]` (Microsoft.Storage/storageAccounts)

The main area displays the JSON template code, with the following visible lines:

```
1 {
2   "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "location": {
6       "type": "string"
7     },
8     "storageAccountName": {
9       "type": "string"
10    },
11    "accountType": {
```

A "Copy template" button is located at the bottom left of the template editor area.



# VS Code: Install Bicep Extension

<https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-bicep>

The screenshot shows the Visual Studio Code interface with the Bicep extension installed. The left sidebar displays the 'EXTENSIONS' view with a search bar and a list of installed extensions. The main editor area shows the 'Bicep' extension details page, including its icon, name, publisher (Microsoft), version (0.33.93), and a list of key features. The 'Installation' section on the right provides details about the extension's identifier, version, last update, and size.

**EXTENSIONS**

Search Extensions in Market...

**INSTALLED** (56)

- .NET Install Tool** (63ms) | Microsoft
- ARM Template Viewer** | Ben Coleman
- Azure Resource Man...** (159ms) | Microsoft
- Azure Resources** | Microsoft
- Bicep** | Microsoft
- C#** | Microsoft

**RECOMMENDED** (0)

**Bicep**

Microsoft | 1,097,596 | ★★★★★ (26)

Bicep language support for Visual Studio Code

Disable | Uninstall | Auto Update

**Key features of the Bicep VS Code extension**

The Bicep VS Code extension is capable of many of the features you would expect out of other language tooling. Here is a comprehensive list of the features that are currently implemented.

**Validation**

The bicep compiler validates that your code is authored correctly. We always validate the syntax of each file and whenever possible also validate the return types of all expressions (functions, resource bodies, parameters, outputs, etc.).

**Installation**

Identifier	ms-azuretools.vscode-bicep
Version	0.33.93
Last Updated	2025-01-31, 11:40:57
Size	99.33MB

**Marketplace**

Published	2020-11-11, 17:32:16
-----------	----------------------

# VS Code: Install ARM Template Extension

<https://marketplace.visualstudio.com/items?itemName=msazurermttools.azurearm-vscode-tools>

The screenshot shows the Visual Studio Code interface with the Azure Resource Manager (ARM) Tools extension installed. The left sidebar displays the 'EXTENSIONS' view with a list of installed extensions, including .NET Install Tool, ARM Template Viewer, Azure Resource Manager Tools (selected), Azure Resources, Bicep, and C#. The main editor area shows the details for the 'Azure Resource Manager (ARM) Tools' extension, which is published by Microsoft. The extension is currently in 'Preview' mode. The details page includes a description, a star rating, and buttons to 'Disable', 'Uninstall', and 'Auto Update'. Below the main description, there are tabs for 'DETAILS', 'FEATURES', 'CHANGELOG', and 'DEPENDENCIES'. The 'Installation' section shows the identifier 'msazurermttools.azurearm-vscode-tools', version '0.15.14', and a last update date of '2024-12-04'. The 'Marketplace' section shows the published date '2016-02-29'. An 'Important notice' section at the bottom states: 'The ARM Tools extension is now in Maintenance Mode. That means that we will...'

EXTENSIONS

Search Extensions in Market...

INSTALLED 56

- .NET Install Tool  
This extension installs and ma...  
Microsoft
- ARM Template Viewer  
Graphically display ARM templ...  
Ben Coleman
- Azure Resource Man...** 108ms  
Language server, editing tools ...  
Microsoft
- Azure Resources  
An extension for viewing and ...  
Microsoft
- Bicep  
Bicep language support for Vis...  
Microsoft
- C#  
Base language support for C#  
Microsoft

RECOMMENDED 0

storage-account.json 5

Extension: Azure Resource Manager (ARM) Tools X

## Azure Resource Manager (ARM) Tools Preview

Microsoft [microsoft.com](https://microsoft.com) | 1,379,862 | ★★★★★ (26)

Language server, editing tools and snippets for Azure Resource Manager (ARM)...

Disable Uninstall ☒ Auto Update

DETAILS FEATURES CHANGELOG DEPENDENCIES

### Azure Resource Manager (ARM) Tools for Visual Studio Code (Preview)

version rate limited by upstream service

vs marketplace rate limited by upstream service

Azure Pipelines succeeded

#### Important notice

The ARM Tools extension is now in Maintenance Mode. That means that we will...

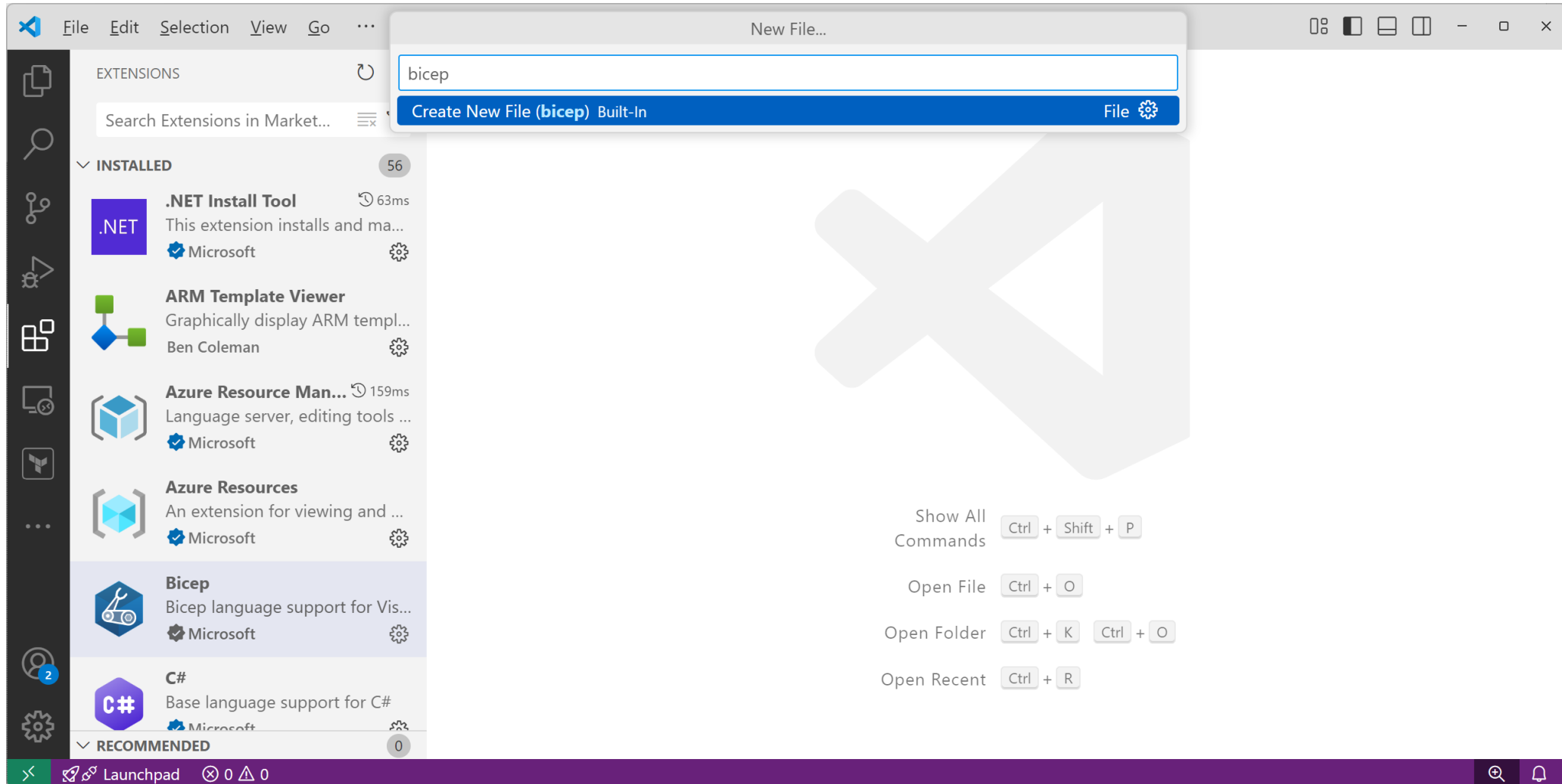
#### Installation

Identifier	msazurermttools.a: vscode-tools
Version	0.15.14
Last Updated	2024-12-04, 15:45:03
Size	49.51MB

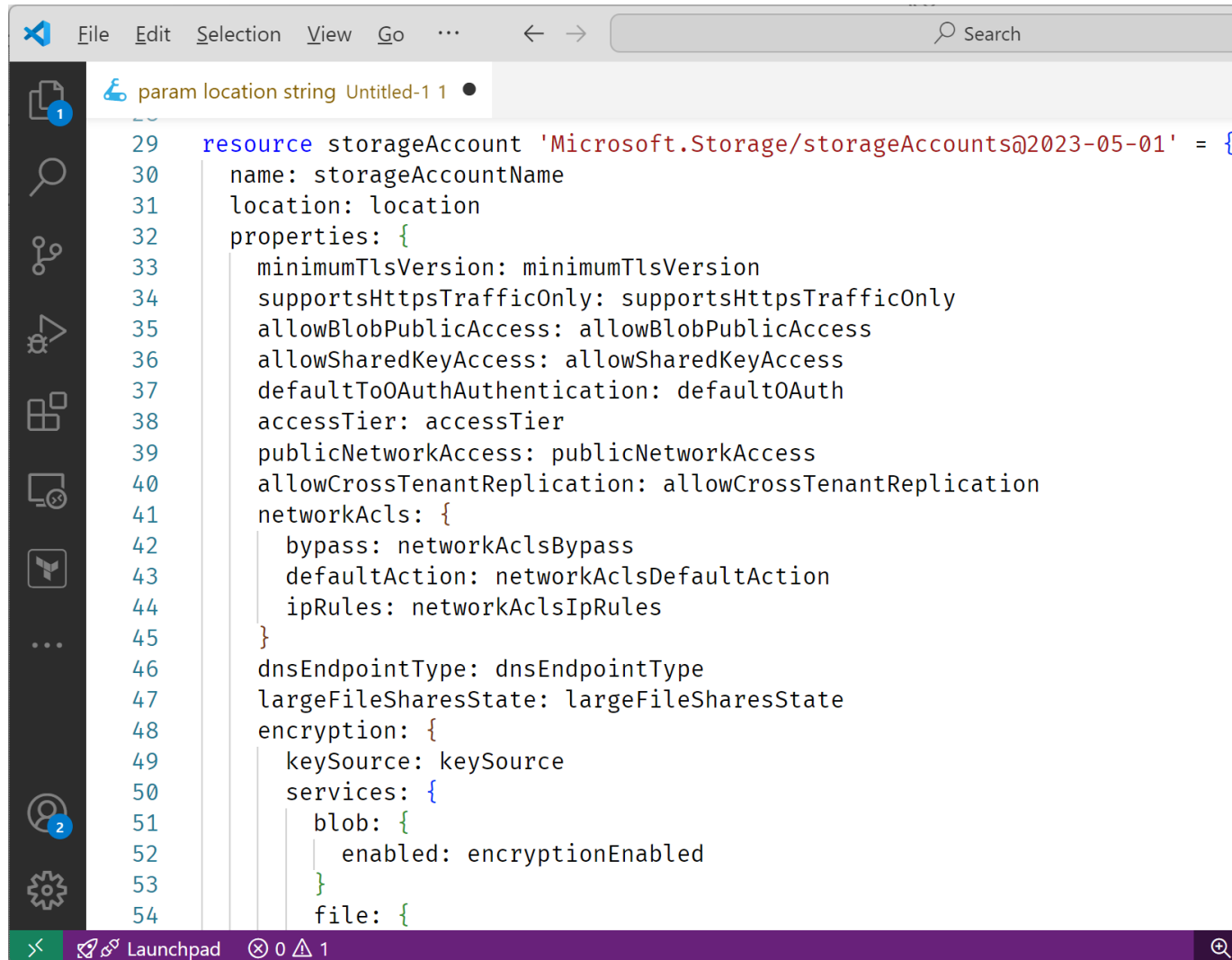
#### Marketplace

Published	2016-02-29, 11:45:34
Last	2024-12-02

# VS Code: New Bicep File



# VS Code: Paste ARM Template JSON



The screenshot shows the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, and a search bar. The left sidebar shows the Explorer, Search, and Run and Debug views. The main editor area displays a file named 'param location string Untitled-1 1'. The code is an ARM Template JSON snippet for a storage account resource. The code is as follows:

```
29 resource storageAccount 'Microsoft.Storage/storageAccounts@2023-05-01' = {  
30   name: storageAccountName  
31   location: location  
32   properties: {  
33     minimumTlsVersion: minimumTlsVersion  
34     supportsHttpsTrafficOnly: supportsHttpsTrafficOnly  
35     allowBlobPublicAccess: allowBlobPublicAccess  
36     allowSharedKeyAccess: allowSharedKeyAccess  
37     defaultToOAuthAuthentication: defaultOAuth  
38     accessTier: accessTier  
39     publicNetworkAccess: publicNetworkAccess  
40     allowCrossTenantReplication: allowCrossTenantReplication  
41     networkAcls: {  
42       bypass: networkAclsBypass  
43       defaultAction: networkAclsDefaultAction  
44       ipRules: networkAclsIpRules  
45     }  
46     dnsEndpointType: dnsEndpointType  
47     largeFileSharesState: largeFileSharesState  
48     encryption: {  
49       keySource: keySource  
50       services: {  
51         blob: {  
52           enabled: encryptionEnabled  
53         }  
54         file: {
```

JSON will be  
decompiled to  
Bicep! 🐱💖

But this doesn't  
make life that much  
easier. 🐱😭

# Why Bicep?

# Azure Bicep

## Objective

- Drastically simplify the authoring experience of Azure resources with a cleaner syntax, improved type safety, and better support for modularity and code re-use.

## Significant Work History

- **2021:**
  - Initial release. Simplified syntax for ARM templates.
  - Enhanced error handling and diagnostic capabilities.
- **2022:**
  - Loops, conditionals, and parameterization.
  - Integration with development tools and CI/CD pipelines.
- **2023:**
  - Bicep Public Module Registry launched.
  - User-defined types added.
- **2024:**
  - User-defined functions added.
  - Public introduction of Azure Verified Modules, pre-configured and rigorously tested building blocks.

## Strengths

- Not JSON, excellent expressivity and modularity support
- Flexible, expressive language and type system
- Active community inside and outside Microsoft
- Some extra provider support: Entra ID, Kubernetes
- Built into Azure, easy learning curve (now), no state files!
- Rich set of patterns and best practices: AVM

## Weaknesses

- Azure + Azure Resource Manager support only
- Hard to validate without deploying (what-if support is iffy)
- Naming things is HARD
- Sometimes annoying documentation (but it's getting better)
- Intellisense for selecting modules from custom registries is not there yet (but accessing the main public registry is not terrible)

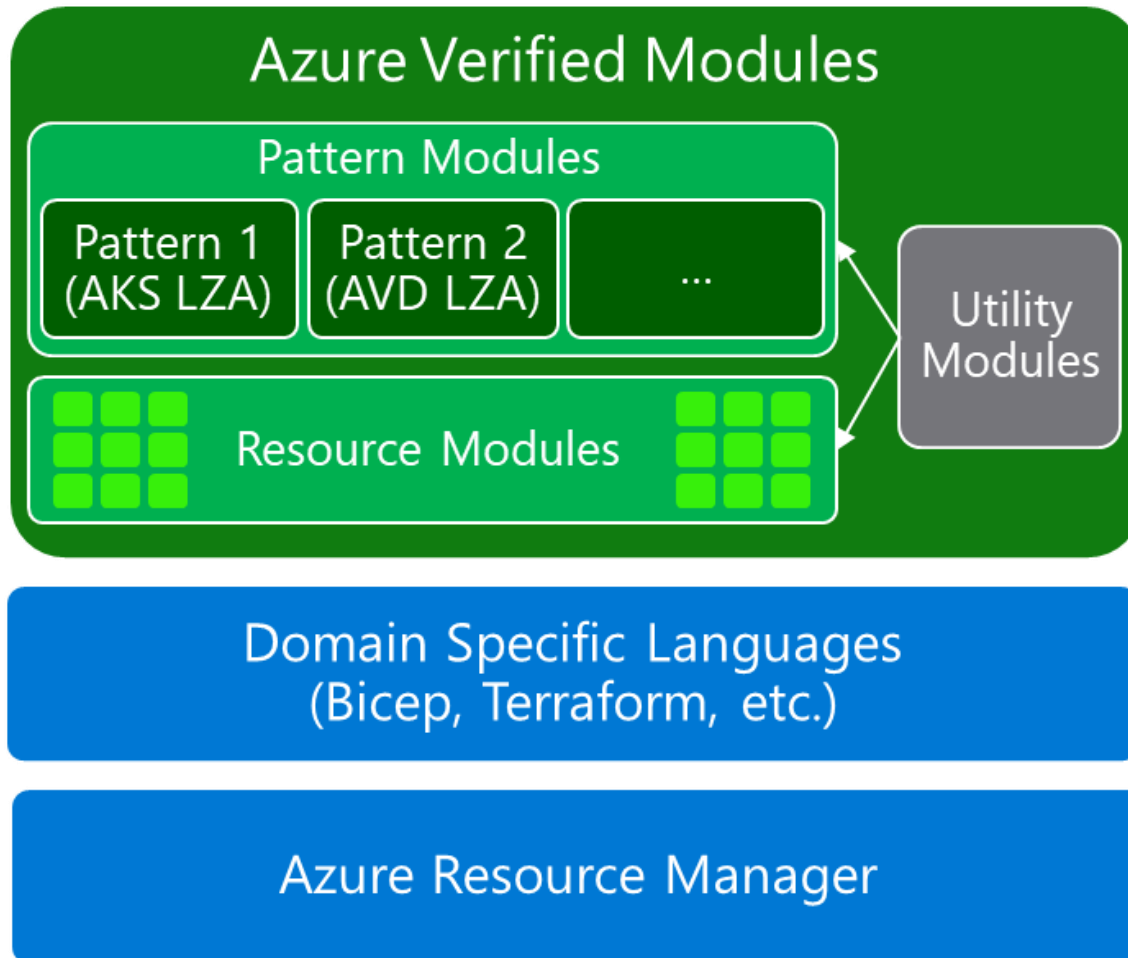
# Expressive Language Features

- User-defined types are super flexible
  - Bicep uses a type system like TypeScript
  - [User-defined types in Bicep - Azure Resource Manager | Microsoft Learn](#)
- Use the import keyword to bring in types and other exported things from outside your Bicep file
  - [Imports in Bicep - Azure Resource Manager | Microsoft Learn](#)
- Find yourself doing the same transformations over and over? Look at user-defined functions
  - [User-defined functions in Bicep - Azure Resource Manager | Microsoft Learn](#)
- Don't repeat yourself, use the built-in functions and lambda expressions to process collections
  - [Bicep functions - lambda - Azure Resource Manager | Microsoft Learn](#)
  - This comes in handy when doing complex things with Application Gateway (you should try to move to FrontDoor instead), or with Virtual Networks
- Safe dereferencing and null-coalescing:
  - [Bicep safe-dereference operator - Azure Resource Manager | Microsoft Learn](#)
  - [Bicep logical operators - Azure Resource Manager | Microsoft Learn](#)

# Modules and Organization



# Using Modules to Organize Infrastructure



**Realm of Laziness**



Work up here!

**Realm of Overwork**



Don't work down here!

<https://azure.github.io/Azure-Verified-Modules/>

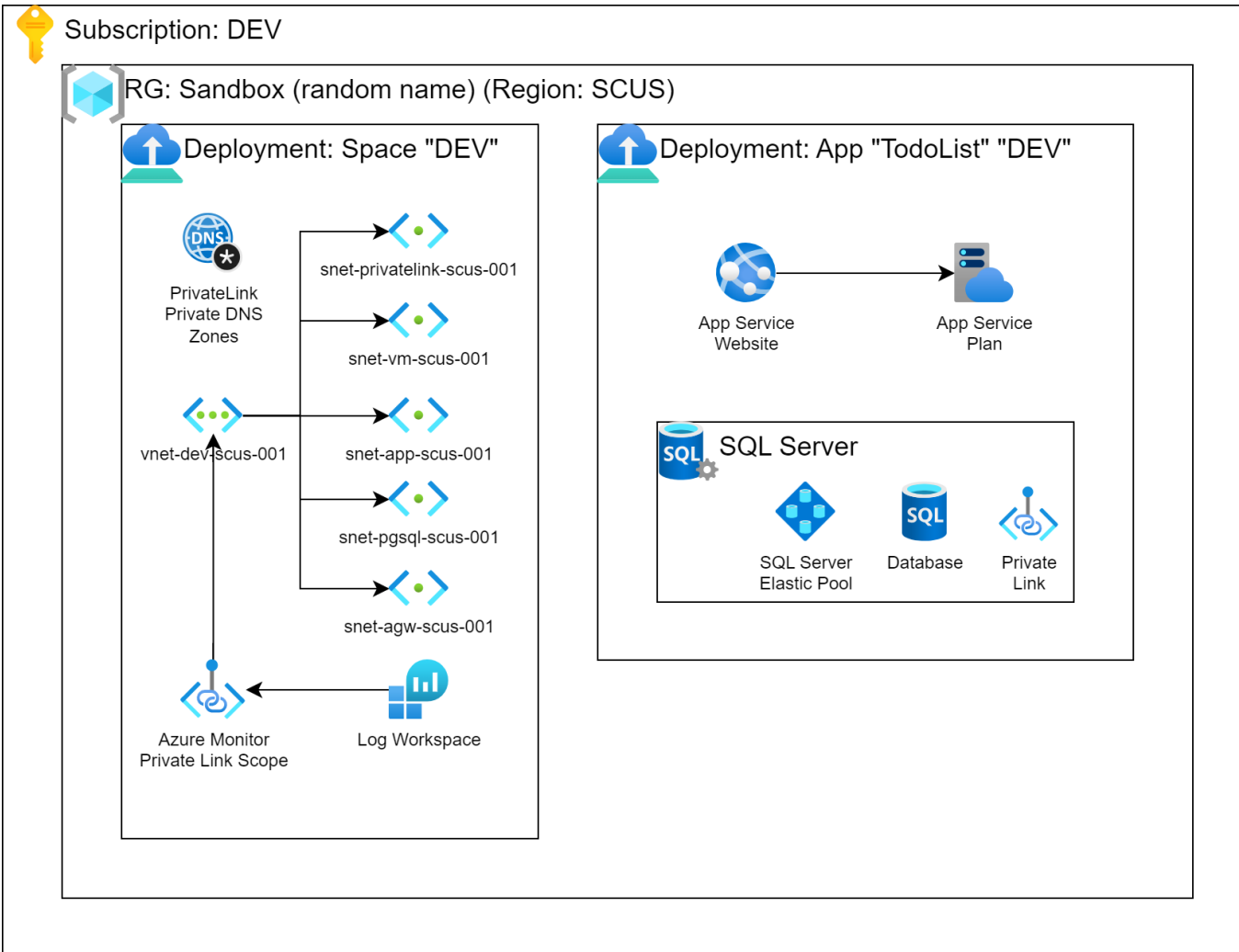
# DEMO

## Azure Verified Modules

# DEMO

## Abstracting the Common Application Patterns

# Cloud Infrastructure



- Use Deployment Stacks to organize resources into managed blocks
- We are cheating a bit due to ACloudGuru's sandbox limitations and using only one Resource Group
- Everything should be kept on the private network, and we can later add a FrontDoor to allow public access to the App Service
- There are some missing integrations like setting up a connection string on the site to the database, and some Key Vault fun

# Modern Infrastructure as Code

# Modern Infrastructure as Code

## Platform Team

- Establish and **Publish** a repository of **patterns** and **best practices**
- Enable **ephemeral** developer sandboxes for **faster iteration** (Azure Deployment Groups / Stacks)

## Development Team

- Use **declarative notation** for choosing infrastructure patterns (Bicep, Score, Terraform, whatever)
- **Talk** to platform team for requesting new infrastructure capabilities

# Naming Things is Hard: You need a **STRATEGY**

- Example: [Develop your naming and tagging strategy for Azure resources - Cloud Adoption Framework | Microsoft Learn](#)
- Be picky on the name structures because once you start it's hard to fix (there's usually no way to rename things)
- If you can, enforce the naming strategy to avoid people making bad choices
- Be explicit about naming by using exported Bicep functions, types, and variables
- Make it easy to adopt the strategy and predict the names of things
- Follow the rules for naming: [Naming rules and restrictions for Azure resources - Azure Resource Manager | Microsoft Learn](#)
- Also: establish a clear tagging strategy and follow it!

# Figuring out Errors SUCKS.

Deploy Workload  
workload.yaml

Filter workflow runs

13 workflow runs

Event Status Branch Actor

This workflow has a workflow\_dispatch event trigger. Run workflow

Deploy Workload	main	1 minute ago	In progress
Deploy Workload #14: Manually run by garoyeri			
Deploy Workload	main	6 minutes ago	2m 47s
Deploy Workload #13: Manually run by garoyeri			
Deploy Workload	main	14 minutes ago	3m 2s
Deploy Workload #12: Manually run by garoyeri			
Deploy Workload	main	21 minutes ago	2m 52s
Deploy Workload #11: Manually run by garoyeri			
Deploy Workload	main	28 minutes ago	4m 13s
Deploy Workload #10: Manually run by garoyeri			

Validating each change takes about 3-5 minutes and I get ONE error message each time.





# New Deployment Groups (Stacks?) Feature

Home > Space-dev Deployment stack

Search

Delete stack Edit and redeploy Refresh

Overview

Access control (IAM)

Settings

Inputs

Outputs

Deployment

Deny settings

Essentials

Subscription: [P5-Real Hands-On Labs](#)

Resource group: [1-7e0e4f03-playground-sandbox](#)

Status: ✓ Succeeded

Last modified: Feb 17, 2025, 11:07 PM

Description: Space Deployment dev

Resources

Filter by resource name or status...

Name	Type	Resource group	Subscription	Deny status
<a href="#">vnet-dev-eus-001/snet-agw-eus-001</a>	Microsoft.Network/virtu...	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none
<a href="#">vnet-dev-eus-001/snet-pgsql-eus-001</a>	Microsoft.Network/virtu...	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none
<a href="#">vnet-dev-eus-001/snet-app-eus-001</a>	Microsoft.Network/virtu...	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none
<a href="#">vnet-dev-eus-001/snet-vm-eus-001</a>	Microsoft.Network/virtu...	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none
<a href="#">vnet-dev-eus-001/snet-privatelink-eus-001</a>	Microsoft.Network/virtu...	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none
<a href="#">vnet-dev-eus-001</a>	Virtual network	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none
<a href="#">log-dev-eus-001</a>	Log Analytics workspace	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none
<a href="#">pls-dev-eus-001</a>	Azure Monitor Private Li...	<a href="#">1-7e0e4f03-pla...</a>	<a href="#">P5-Real Hands-On Labs</a>	none

- Does not support “whatif” but that’s probably ok (it wasn’t always that helpful)
- Allows for deletion of resources that don’t appear in Bicep scripts (or detach)
- Can add “deny” rules to block deletion of stacks until “deny” block is removed (doesn’t replace “locks” though).
- Great for developer iterations: bring up an environment, test it, tear it down without bothering others (deleting a stack deletes those resources).

# Next Steps

- These examples are an OK starting point to get you familiar with how to use Bicep. Your actual Azure environment will be different (this demo has limitations due to the sandbox provided by ACloudGuru).
- Secrets management is a big concern: there are some automations you can do with Key Vault or consider a different 3<sup>rd</sup> party service with APIs and stuff.
  - You can roll your own secrets rotation with Key Vault and GitHub Actions, but it requires a lot of manual work
  - Hashicorp Vault and CyberArk Conjur provide a framework for building best practices around secrets rotation (but can be complicated to setup correctly)
- Establish a Platform Team who can focus on this stuff. It's HARD and COMPLICATED and FRUSTRATING.
  - Give them time and money
  - Consider 3<sup>rd</sup> party tools like [Harness](#) and Pulumi that give you more complete solutions over plain Bicep + ARM
  - Allow teams some bandwidth to fail and experiment to get to something that works for you
- Consider joining the Bicep Community Calls: <https://aka.ms/armnews> (they are super friendly and are showing new stuff each month)
  - Watch for AVM Community Calls: [Community Calls | AVM](#)



**Thank You**

