Give enterprise wide 'managed' freedom with Azure Template specs with Bicep

Mónika Nagy & Thomas Voorend

Rabobank

What are we doing as Platform Engineers?

3 teams with all
8 platform engineers

to enable Azure
within Rabobank

1208 subscriptions
for development
workloads

1002 subscriptions
for production
workloads

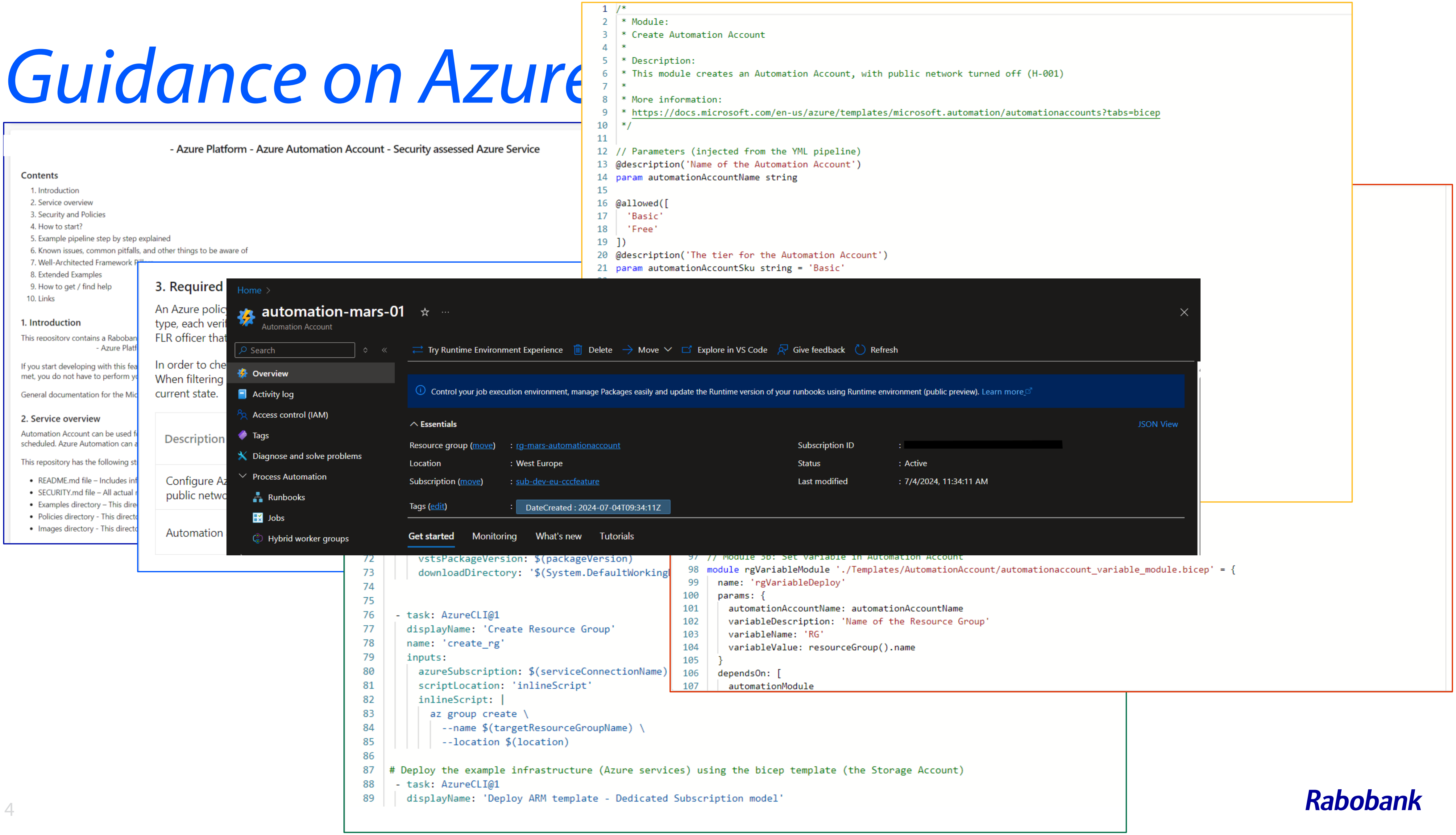
245.000 resources
within Azure

Rabobank

How to guide teams to use Azure?

Rabobank

Guidance on Azure



- Azure Platform - Azure Automation Account - Security assessed Azure Service

Contents

1. Introduction

This repository contains a Raboban
- Azure Platf

If you start developing with this fea
met, you do not have to perform yo

General documentation for the Mic

2. Service overview

Automation Account can be used f
scheduled. Azure Automation can a

This repository has the following st

- README.md file – Includes inf
- SECURITY.md file – All actual r
- Examples directory – This dire
- Policies directory - This directo
- Images directory - This directo

3. Required

An Azure polic
type, each veri
FLR officer that

In order to che
When filtering
current state.

Description

Configure Az
public netwo

Automation

```
/*
 * Module:
 * Create Automation Account
 *
 * Description:
 * This module creates an Automation Account, with public network turned off (H-001)
 *
 * More information:
 * https://docs.microsoft.com/en-us/azure/templates/microsoft.automation/automationaccounts?tabs=bicep
 */

// Parameters (injected from the YML pipeline)
@description('Name of the Automation Account')
param automationAccountName string

@allowed([
  'Basic'
  'Free'
])
@description('The tier for the Automation Account')
param automationAccountSku string = 'Basic'
```

Home >

automation-mars-01

Automation Account

Control your job execution environment, manage Packages easily and update the Runtime version of your runbooks using Runtime environment (public preview). Learn more

Essentials

JSON View

Resource group (move) : rg-mars-automationaccount
Location : West Europe
Subscription (move) : sub-dev-eu-cccfeature
Tags (edit) : DateCreated : 2024-07-04T09:34:11Z

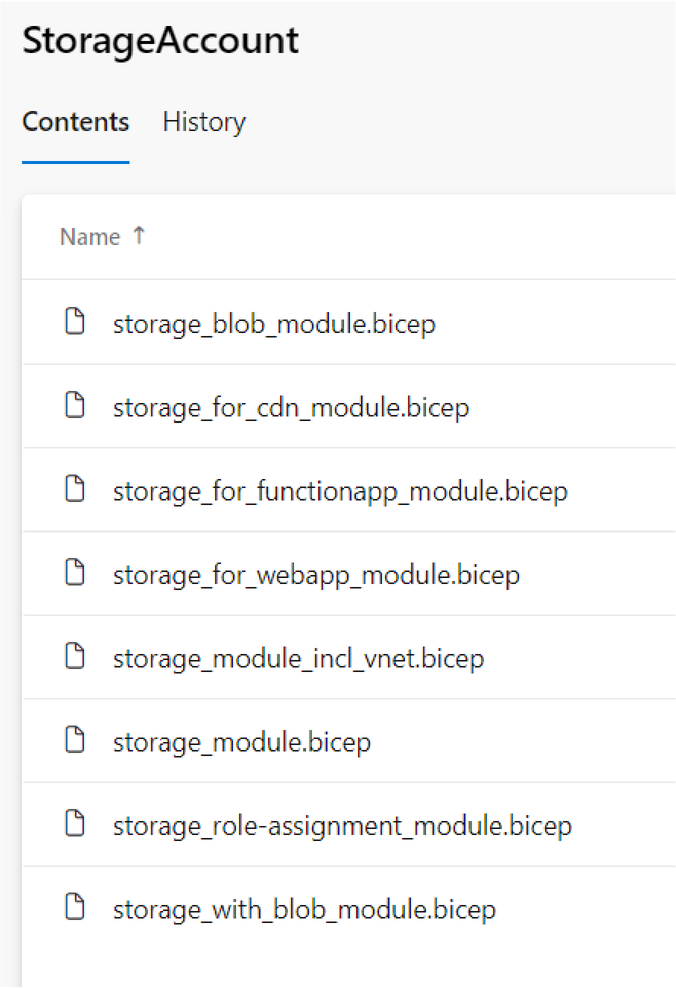
Subscription ID : [redacted]
Status : Active
Last modified : 7/4/2024, 11:34:11 AM

```
      vstsPackageVersion: $(packageVersion)
      downloadDirectory: '$(System.DefaultWorking


  - task: AzureCLI@1
    displayName: 'Create Resource Group'
    name: 'create_rg'
    inputs:
      azureSubscription: $(serviceConnectionName)
      scriptLocation: 'inlineScript'
      inlineScript: |
        az group create \
          --name $(targetResourceGroupName) \
          --location $(location)

# Deploy the example infrastructure (Azure services) using the bicep template (the Storage Account)
  - task: AzureCLI@1
    displayName: 'Deploy ARM template - Dedicated Subscription model'
```

```
// Module 3b: Set variable in Automation Account
module rgVariableModule './Templates/AutomationAccount/automationaccount_variable_module.bicep' = {
  name: 'rgVariableDeploy'
  params: {
    automationAccountName: automationAccountName
    variableDescription: 'Name of the Resource Group'
    variableName: 'RG'
    variableValue: resourceGroup().name
  }
  dependsOn: [
    automationModule
```

Rabobank

Multiple modules to rule them all?



StorageAccount

Name ↑

- storage_blob_module.bicep
- storage_for_cdn_module.bicep
- storage_for_functionapp_module.bicep
- storage_for_webapp_module.bicep
- storage_module_incl_vnet.bicep
- storage_module.bicep
- storage_role-assignment_module.bicep
- storage_with_blob_module.bicep

Guidance on Azure

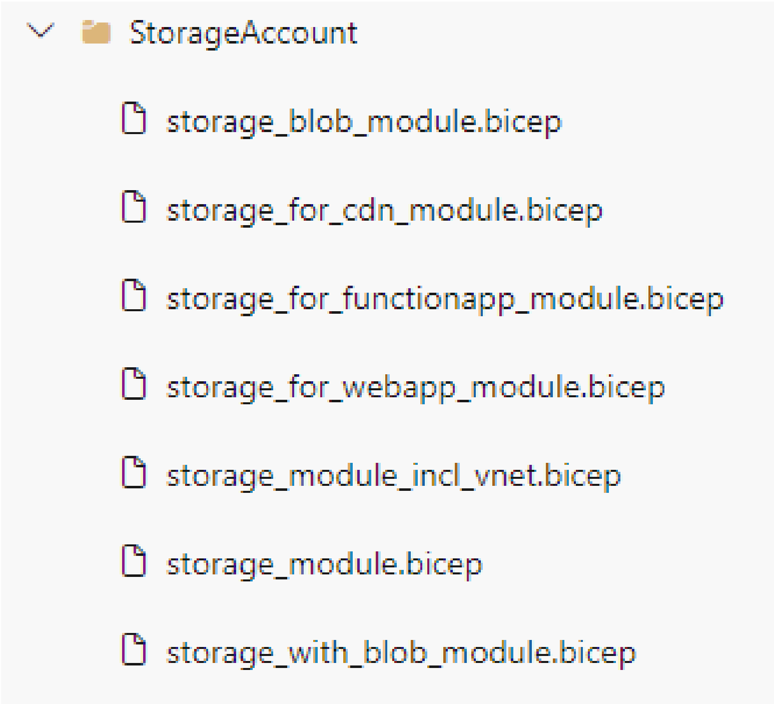
Compliant Azure Service for teams to deploy

What about multiple resources?

Cognitive load?

Can this be less?

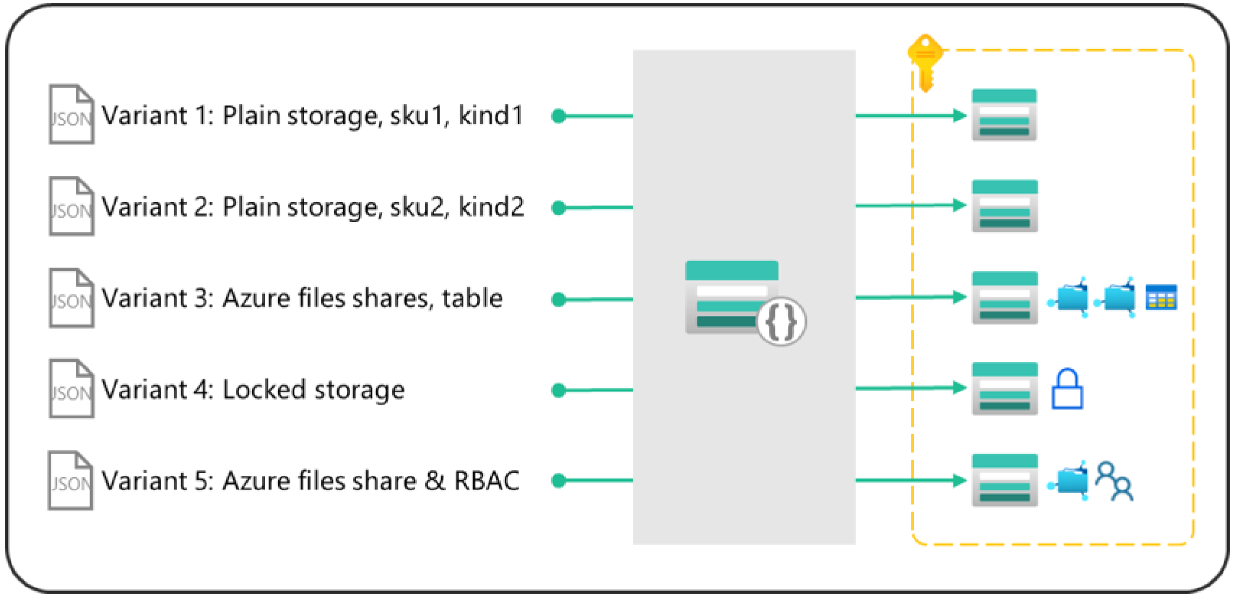
A new approach: standardized templates!



StorageAccount

- storage_blob_module.bicep
- storage_for_cdn_module.bicep
- storage_for_functionapp_module.bicep
- storage_for_webapp_module.bicep
- storage_module_incl_vnet.bicep
- storage_module.bicep
- storage_with_blob_module.bicep

Template
per use case



Variant 1: Plain storage, sku1, kind1

Variant 2: Plain storage, sku2, kind2

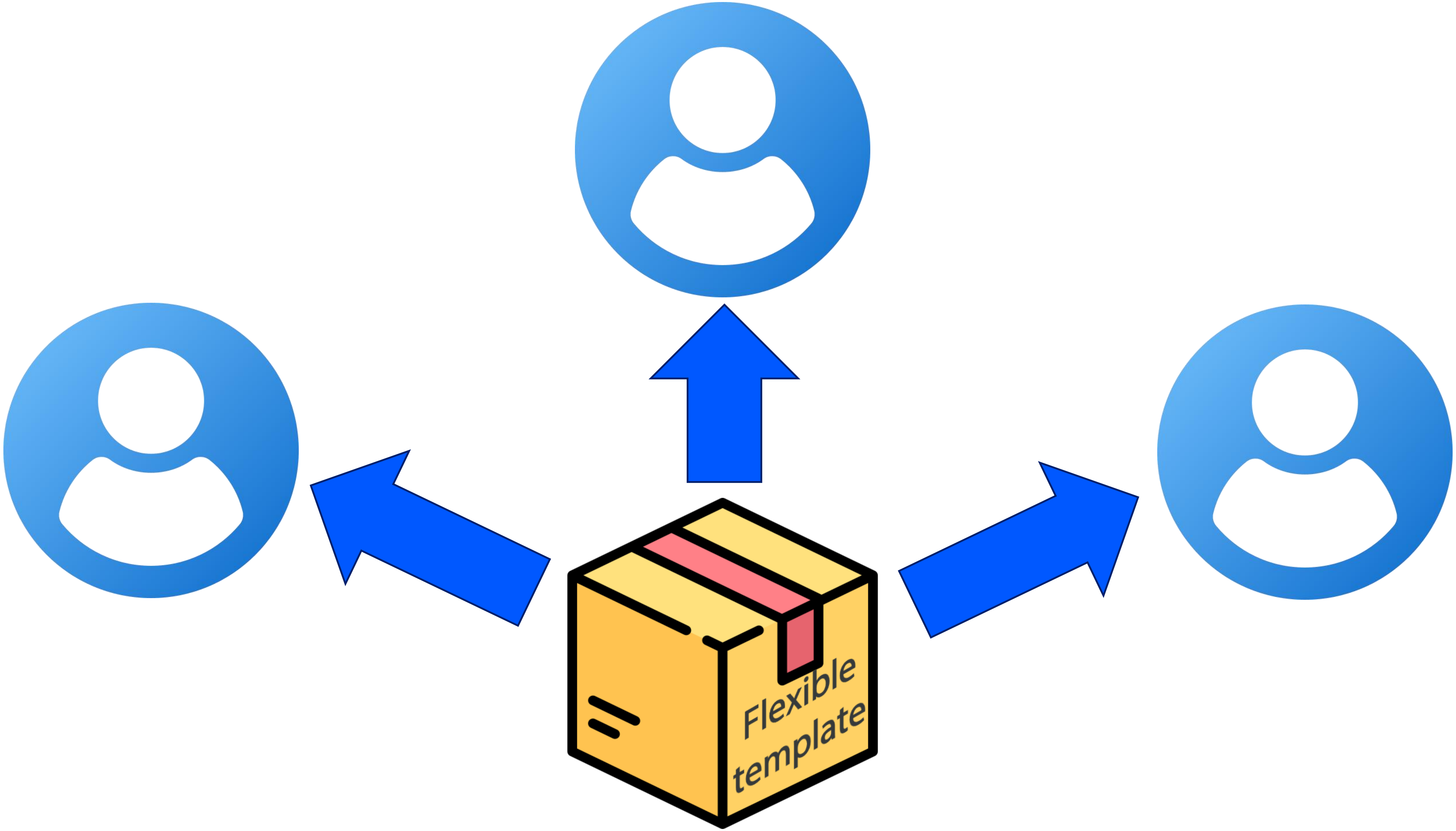
Variant 3: Azure files shares, table

Variant 4: Locked storage

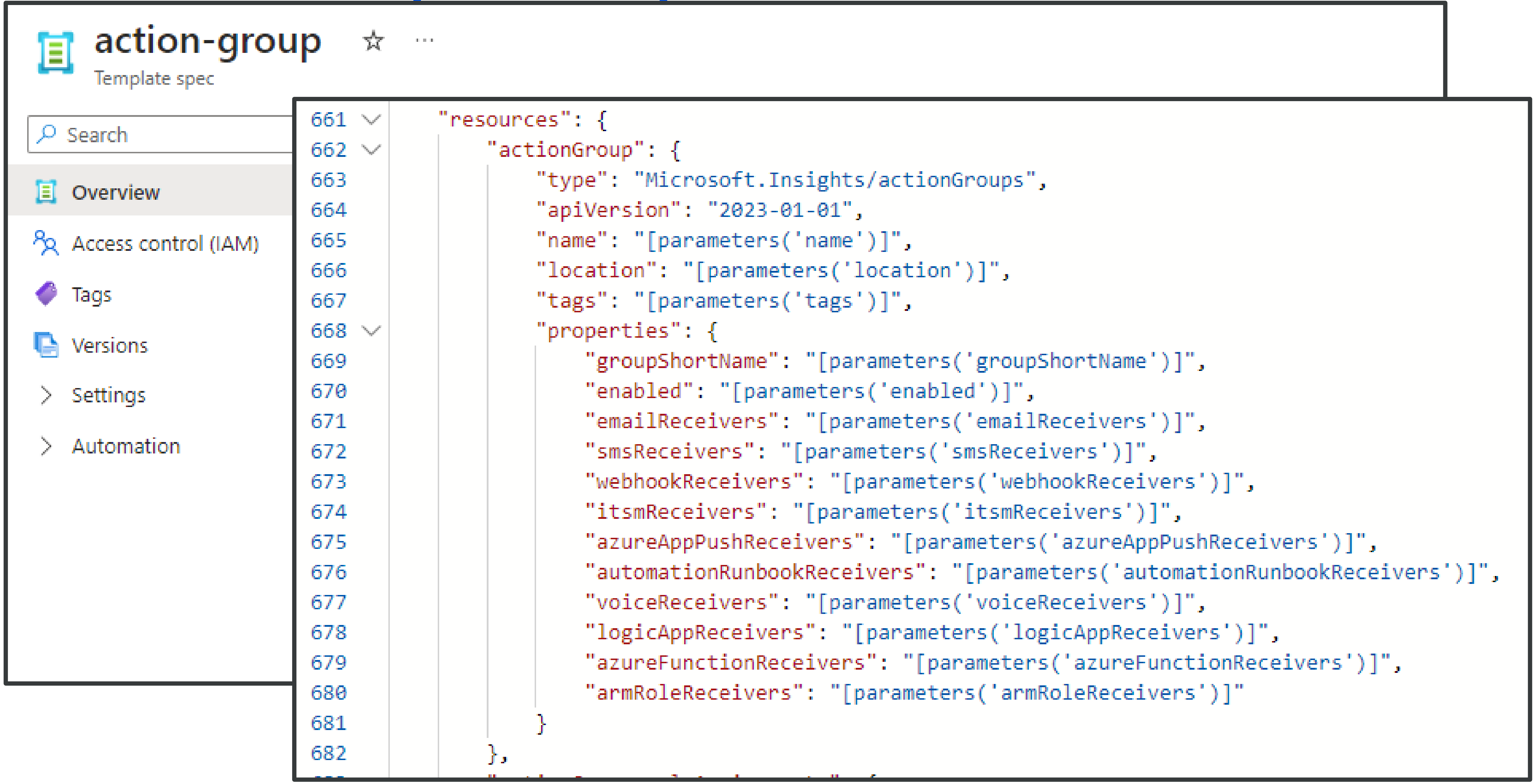
Variant 5: Azure files share & RBAC

Parameter file
per use case

One template

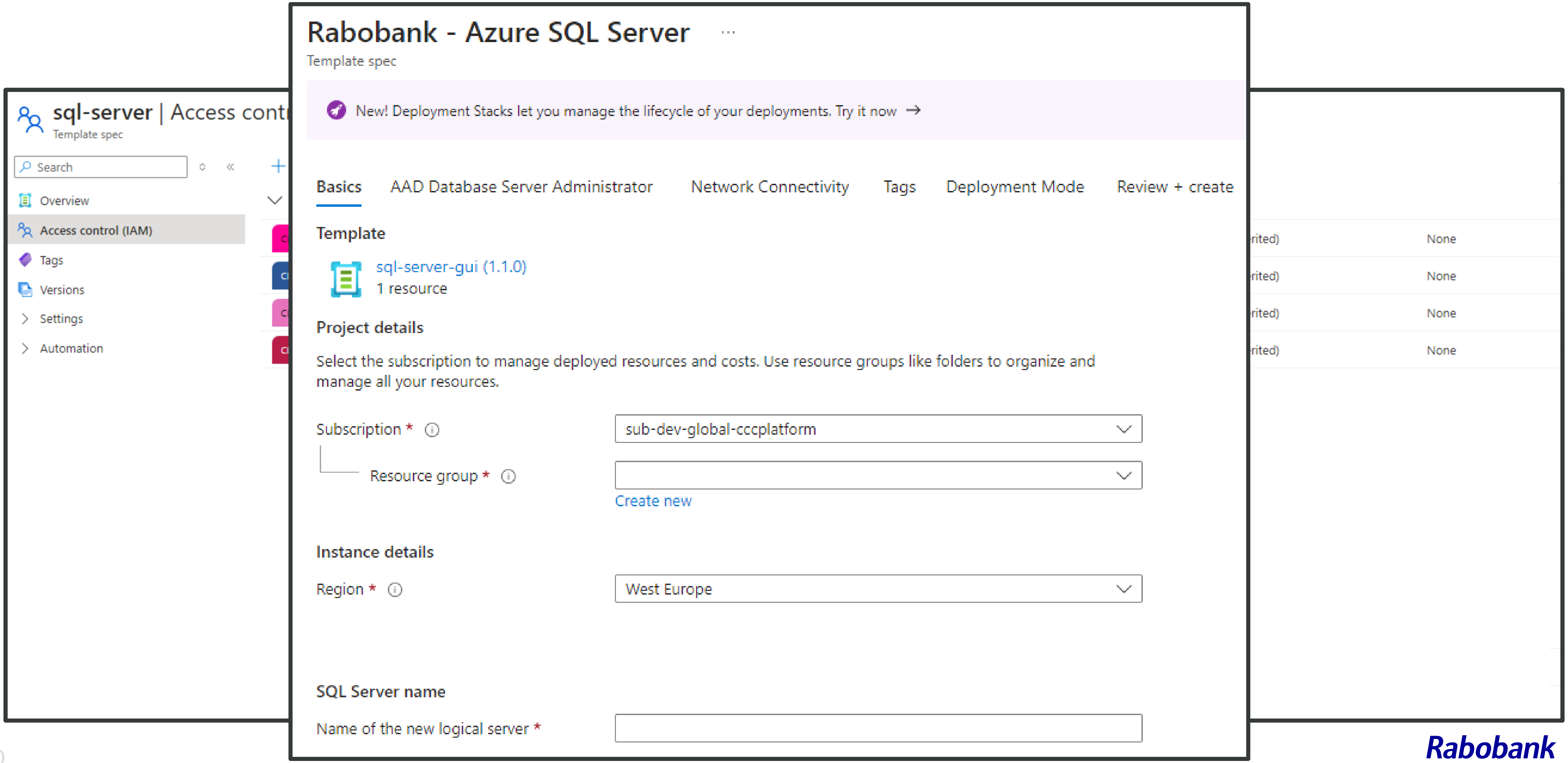

Flexible template

Azure Template Specs



```
    "resources": {
        "actionGroup": {
            "type": "Microsoft.Insights/actionGroups",
            "apiVersion": "2023-01-01",
            "name": "[parameters('name')]",
            "location": "[parameters('location')]",
            "tags": "[parameters('tags')]",
            "properties": {
                "groupShortName": "[parameters('groupShortName')]",
                "enabled": "[parameters('enabled')]",
                "emailReceivers": "[parameters('emailReceivers')]",
                "smsReceivers": "[parameters('smsReceivers')]",
                "webhookReceivers": "[parameters('webhookReceivers')]",
                "itsmReceivers": "[parameters('itsmReceivers')]",
                "azureAppPushReceivers": "[parameters('azureAppPushReceivers')]",
                "automationRunbookReceivers": "[parameters('automationRunbookReceivers')]",
                "voiceReceivers": "[parameters('voiceReceivers')]",
                "logicAppReceivers": "[parameters('logicAppReceivers')]",
                "azureFunctionReceivers": "[parameters('azureFunctionReceivers')]",
                "armRoleReceivers": "[parameters('armRoleReceivers')]"
            }
        },
```

Advantages



Rabobank - Azure SQL Server

Template spec

New! Deployment Stacks let you manage the lifecycle of your deployments. Try it now →

Template

sql-server-gui (1.1.0)
1 resource

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : sub-dev-global-cccplatform

Resource group * :

Create new

Instance details

Region * : West Europe

SQL Server name

Name of the new logical server * :

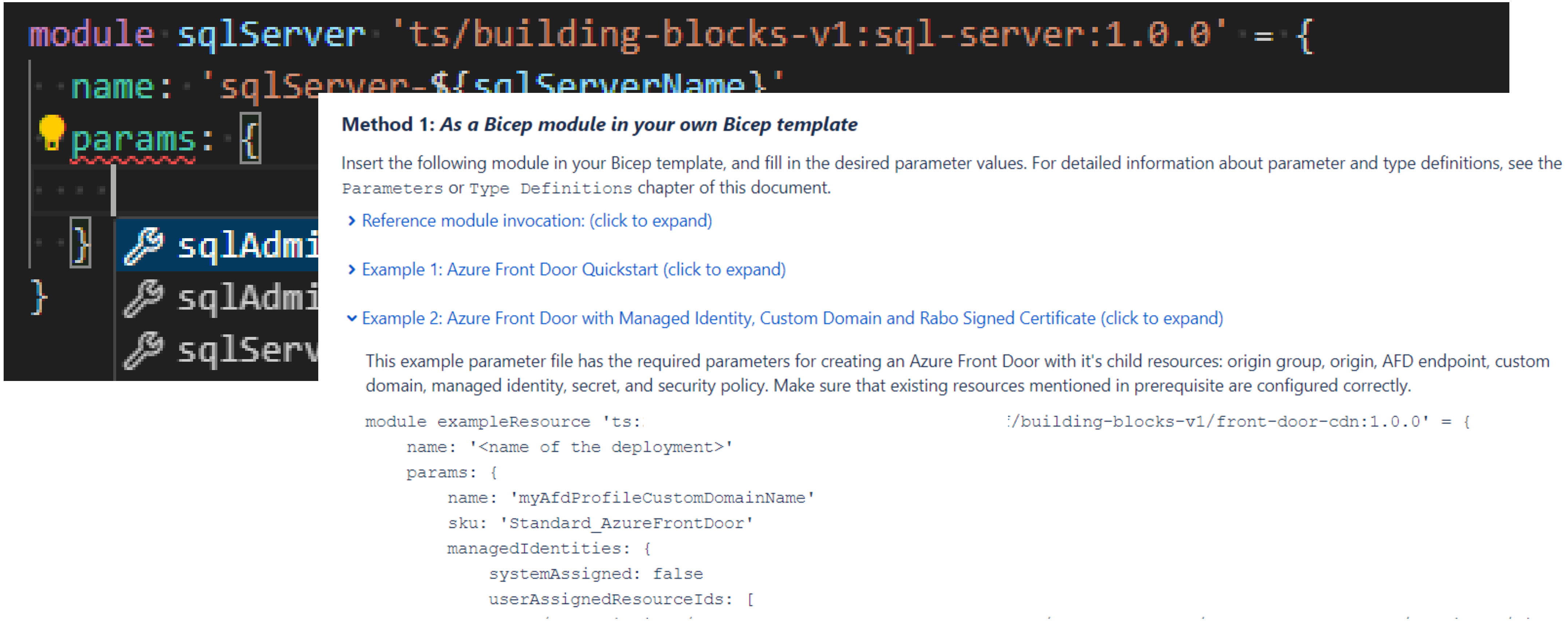
Deploy Template Spec

Deploy Template Spec

In Bicep

```
module exampleResource 'ts/building-blocks-v1:sql-server:1.0.0' = {
  name: 'deploy-sqlserver'
  params: {
      name: ape-sql'
      …
      }
  }
}
```

Parameters



```
module sqlServer 'ts/building-blocks-v1:sql-server:1.0.0' = {
  name: 'sqlServer-${sqlServerName}'
  params: {
    
  }
}
```

Method 1: ***As a Bicep module in your own Bicep template***

Insert the following module in your Bicep template, and fill in the desired parameter values. For detailed information about parameter and type definitions, see the `Parameters` or `Type Definitions` chapter of this document.

› Reference module invocation: (click to expand)

› Example 1: Azure Front Door Quickstart (click to expand)

⌄ Example 2: Azure Front Door with Managed Identity, Custom Domain and Rabo Signed Certificate (click to expand)

This example parameter file has the required parameters for creating an Azure Front Door with it's child resources: origin group, origin, AFD endpoint, custom domain, managed identity, secret, and security policy. Make sure that existing resources mentioned in prerequisite are configured correctly.

```
module exampleResource 'ts:                                    :/building-blocks-v1/front-door-cdn:1.0.0' = {
    name: '<name of the deployment>'
    params: {
        name: 'myAfdProfileCustomDomainName'
        sku: 'Standard_AzureFrontDoor'
        managedIdentities: {
            systemAssigned: false
            userAssignedResourceIds: [
```

Re-invent the wheel?

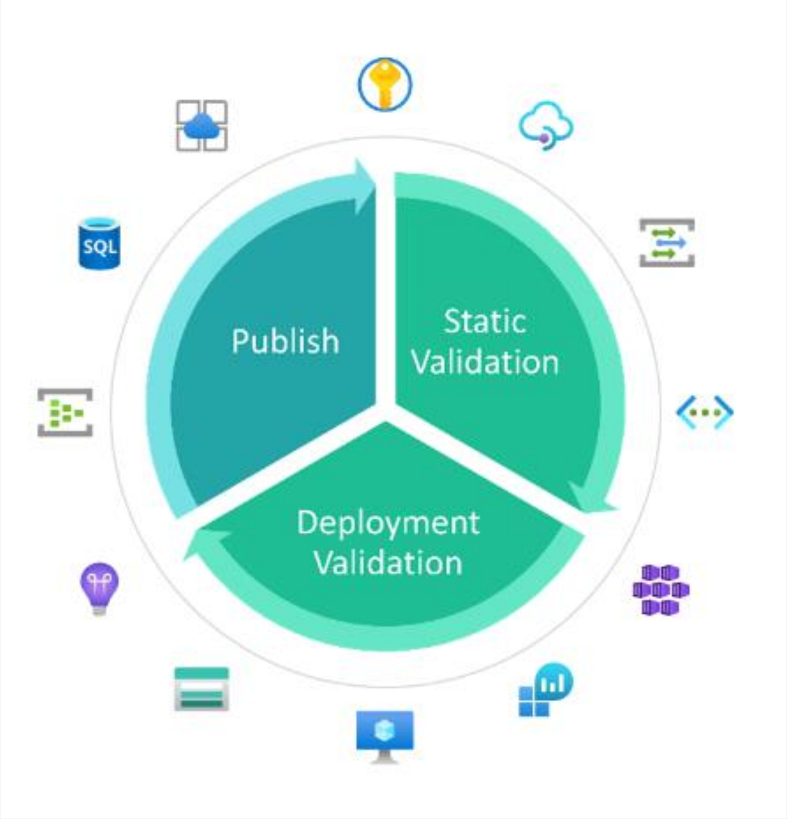
AVM

Value Proposition

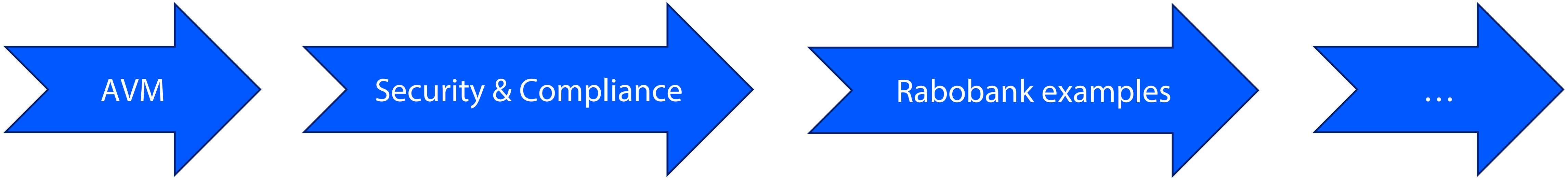
Azure Verified Modules (AVM) is an initiative to consolidate and set the *standards for what a good Infrastructure-as-Code module looks like*.

Modules will then align to these standards, across languages (Bicep, Terraform etc.) and will then be classified as AVMs and available from their respective language specific registries.

AVM is a common code base, a toolkit for our Customers, our Partners, and Microsoft. *It's an official, Microsoft driven initiative*, with a devolved ownership approach to develop modules, leveraging internal & external communities.



How are we developing now?



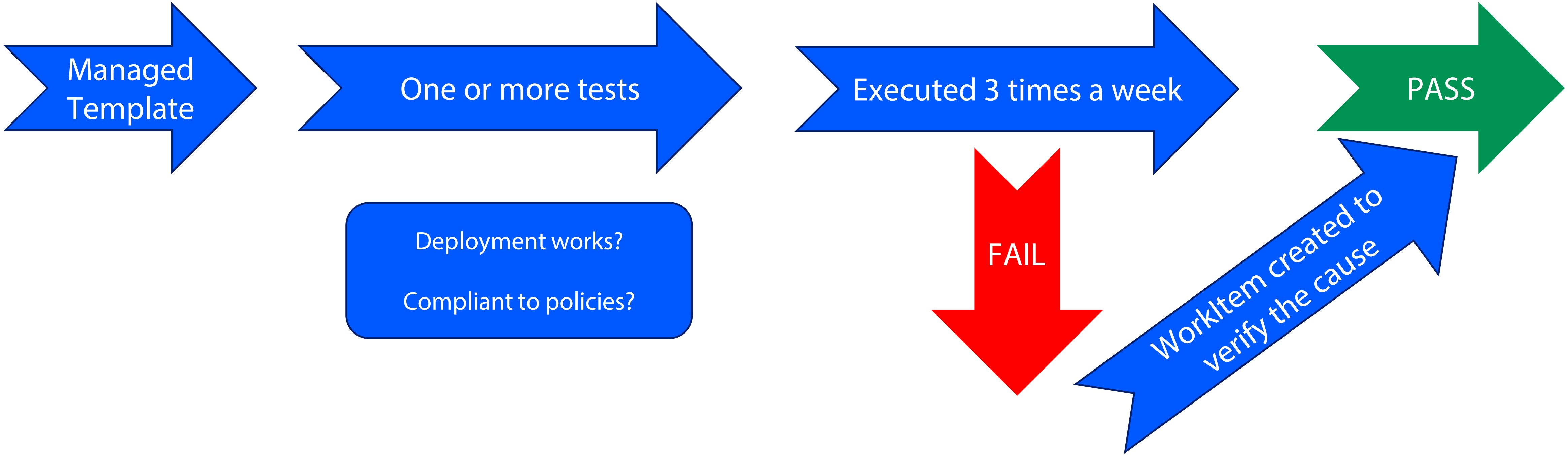
AVM

Security & Compliance

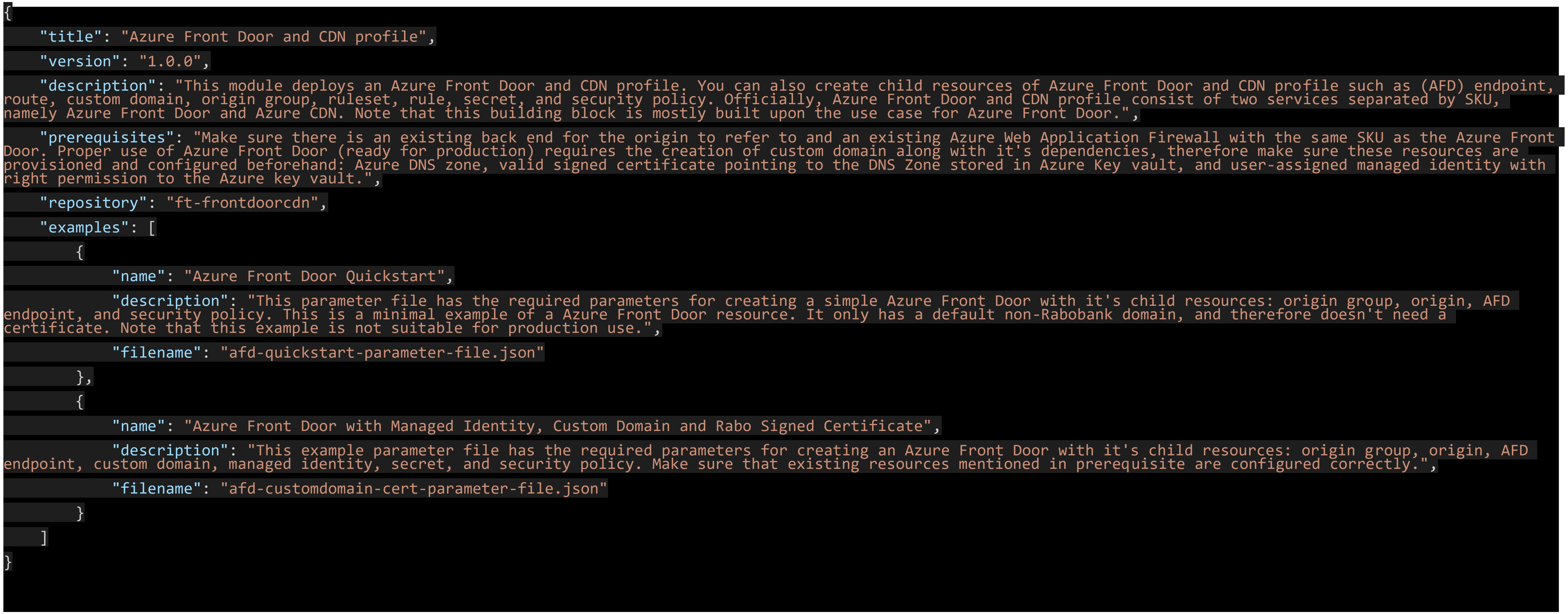
Rabobank examples

…

It's Azure, is everything still working?

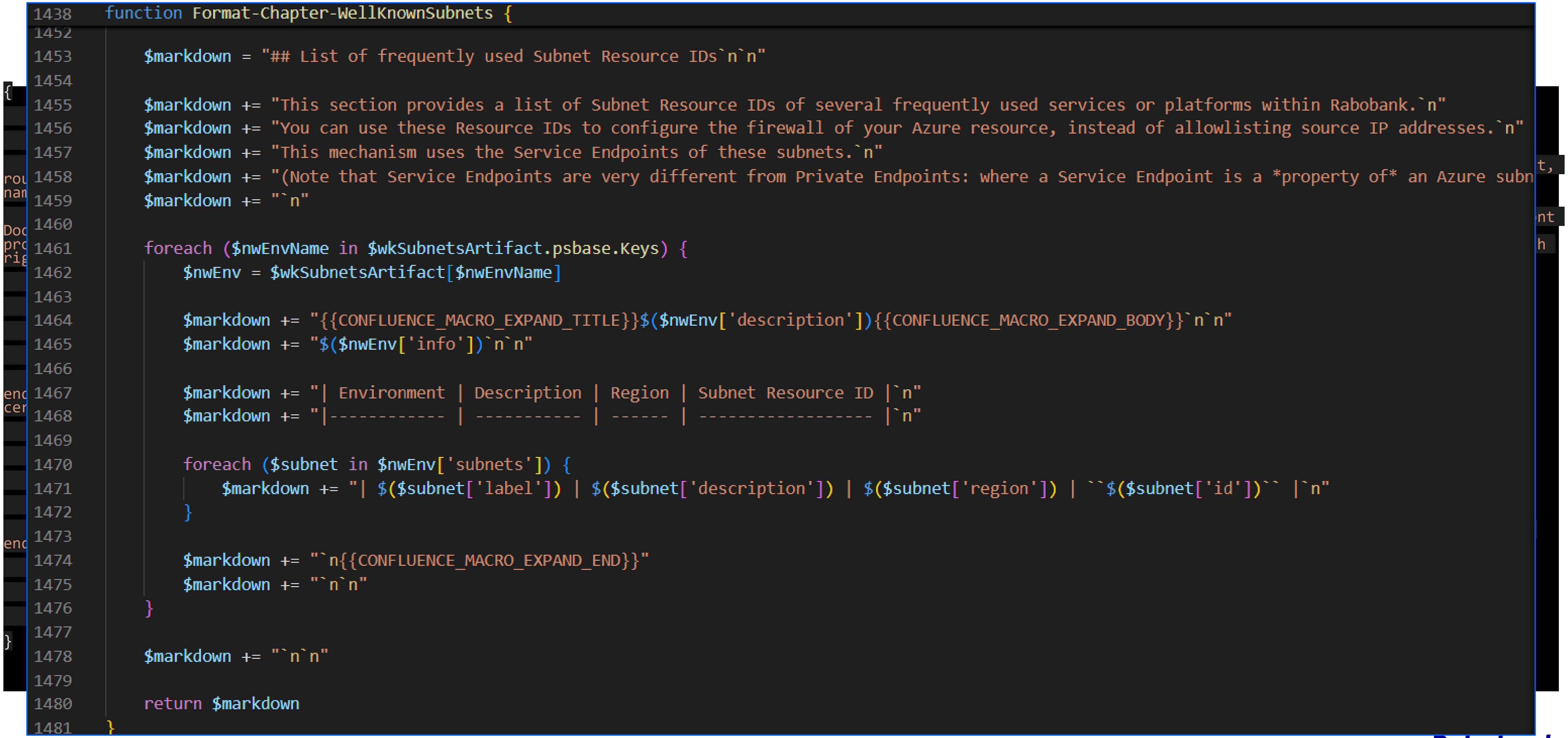


How we document



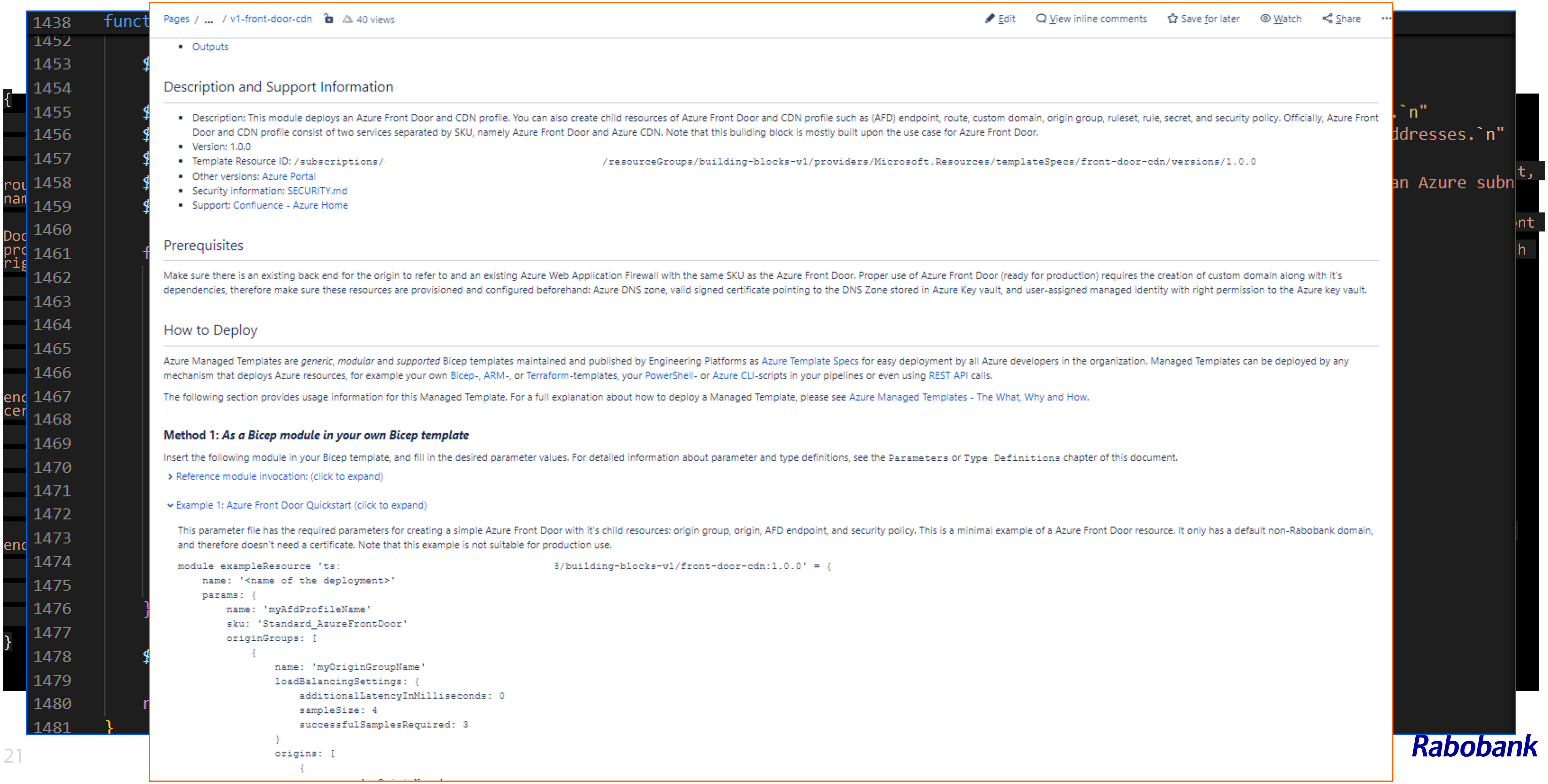
```
{
    "title": "Azure Front Door and CDN profile",
    "version": "1.0.0",
    "description": "This module deploys an Azure Front Door and CDN profile. You can also create child resources of Azure Front Door and CDN profile such as (AFD) endpoint, route, custom domain, origin group, ruleset, rule, secret, and security policy. Officially, Azure Front Door and CDN profile consist of two services separated by SKU, namely Azure Front Door and Azure CDN. Note that this building block is mostly built upon the use case for Azure Front Door.",
    "prerequisites": "Make sure there is an existing back end for the origin to refer to and an existing Azure Web Application Firewall with the same SKU as the Azure Front Door. Proper use of Azure Front Door (ready for production) requires the creation of custom domain along with it's dependencies, therefore make sure these resources are provisioned and configured beforehand: Azure DNS zone, valid signed certificate pointing to the DNS Zone stored in Azure Key vault, and user-assigned managed identity with right permission to the Azure key vault.",
    "repository": "ft-frontdoorcdn",
    "examples": [
        {
            "name": "Azure Front Door Quickstart",
            "description": "This parameter file has the required parameters for creating a simple Azure Front Door with it's child resources: origin group, origin, AFD endpoint, and security policy. This is a minimal example of a Azure Front Door resource. It only has a default non-Rabobank domain, and therefore doesn't need a certificate. Note that this example is not suitable for production use.",
            "filename": "afd-quickstart-parameter-file.json"
        },
        {
            "name": "Azure Front Door with Managed Identity, Custom Domain and Rabo Signed Certificate",
            "description": "This example parameter file has the required parameters for creating an Azure Front Door with it's child resources: origin group, origin, AFD endpoint, custom domain, managed identity, secret, and security policy. Make sure that existing resources mentioned in prerequisite are configured correctly.",
            "filename": "afd-customdomain-cert-parameter-file.json"
        }
    ]
}
```

How we document



```
function Format-Chapter-WellKnownSubnets {

    $markdown = "## List of frequently used Subnet Resource IDs`n`n"

    $markdown += "This section provides a list of Subnet Resource IDs of several frequently used services or platforms within Rabobank.`n"
    $markdown += "You can use these Resource IDs to configure the firewall of your Azure resource, instead of allowlisting source IP addresses.`n"
    $markdown += "This mechanism uses the Service Endpoints of these subnets.`n"
    $markdown += "(Note that Service Endpoints are very different from Private Endpoints: where a Service Endpoint is a *property of* an Azure subn
    $markdown += "`n"

    foreach ($nwEnvName in $wkSubnetsArtifact.psbase.Keys) {
        $nwEnv = $wkSubnetsArtifact[$nwEnvName]

        $markdown += "{{CONFLUENCE_MACRO_EXPAND_TITLE}}$($nwEnv['description']){{CONFLUENCE_MACRO_EXPAND_BODY}}`n`n"
        $markdown += "$($nwEnv['info'])`n`n"

        $markdown += "| Environment | Description | Region | Subnet Resource ID |`n"
        $markdown += "|------------ | ----------- | ------ | ------------------ |`n"

        foreach ($subnet in $nwEnv['subnets']) {
            $markdown += "| $($subnet['label']) | $($subnet['description']) | $($subnet['region']) | ``$($subnet['id'])`` |`n"
        }

        $markdown += "`n{{CONFLUENCE_MACRO_EXPAND_END}}"
        $markdown += "`n`n"
    }

    $markdown += "`n`n"

    return $markdown
}
```

How we document



- Outputs

Description and Support Information

- Description: This module deploys an Azure Front Door and CDN profile. You can also create child resources of Azure Front Door and CDN profile such as (AFD) endpoint, route, custom domain, origin group, ruleset, rule, secret, and security policy. Officially, Azure Front Door and CDN profile consist of two services separated by SKU, namely Azure Front Door and Azure CDN. Note that this building block is mostly built upon the use case for Azure Front Door.
- Version: 1.0.0
- Template Resource ID: `/subscriptions/` `/resourceGroups/building-blocks-v1/providers/Microsoft.Resources/templateSpecs/front-door-cdn/versions/1.0.0`
- Other versions: Azure Portal
- Security information: SECURITY.md
- Support: Confluence - Azure Home

Prerequisites

Make sure there is an existing back end for the origin to refer to and an existing Azure Web Application Firewall with the same SKU as the Azure Front Door. Proper use of Azure Front Door (ready for production) requires the creation of custom domain along with it's dependencies, therefore make sure these resources are provisioned and configured beforehand: Azure DNS zone, valid signed certificate pointing to the DNS Zone stored in Azure Key vault, and user-assigned managed identity with right permission to the Azure key vault.

How to Deploy

Azure Managed Templates are *generic*, *modular* and *supported* Bicep templates maintained and published by Engineering Platforms as Azure Template Specs for easy deployment by all Azure developers in the organization. Managed Templates can be deployed by any mechanism that deploys Azure resources, for example your own Bicep-, ARM-, or Terraform-templates, your PowerShell- or Azure CLI-scripts in your pipelines or even using REST API calls.

The following section provides usage information for this Managed Template. For a full explanation about how to deploy a Managed Template, please see Azure Managed Templates - The What, Why and How.

Method 1: *As a Bicep module in your own Bicep template*

Insert the following module in your Bicep template, and fill in the desired parameter values. For detailed information about parameter and type definitions, see the `Parameters` or `Type Definitions` chapter of this document.

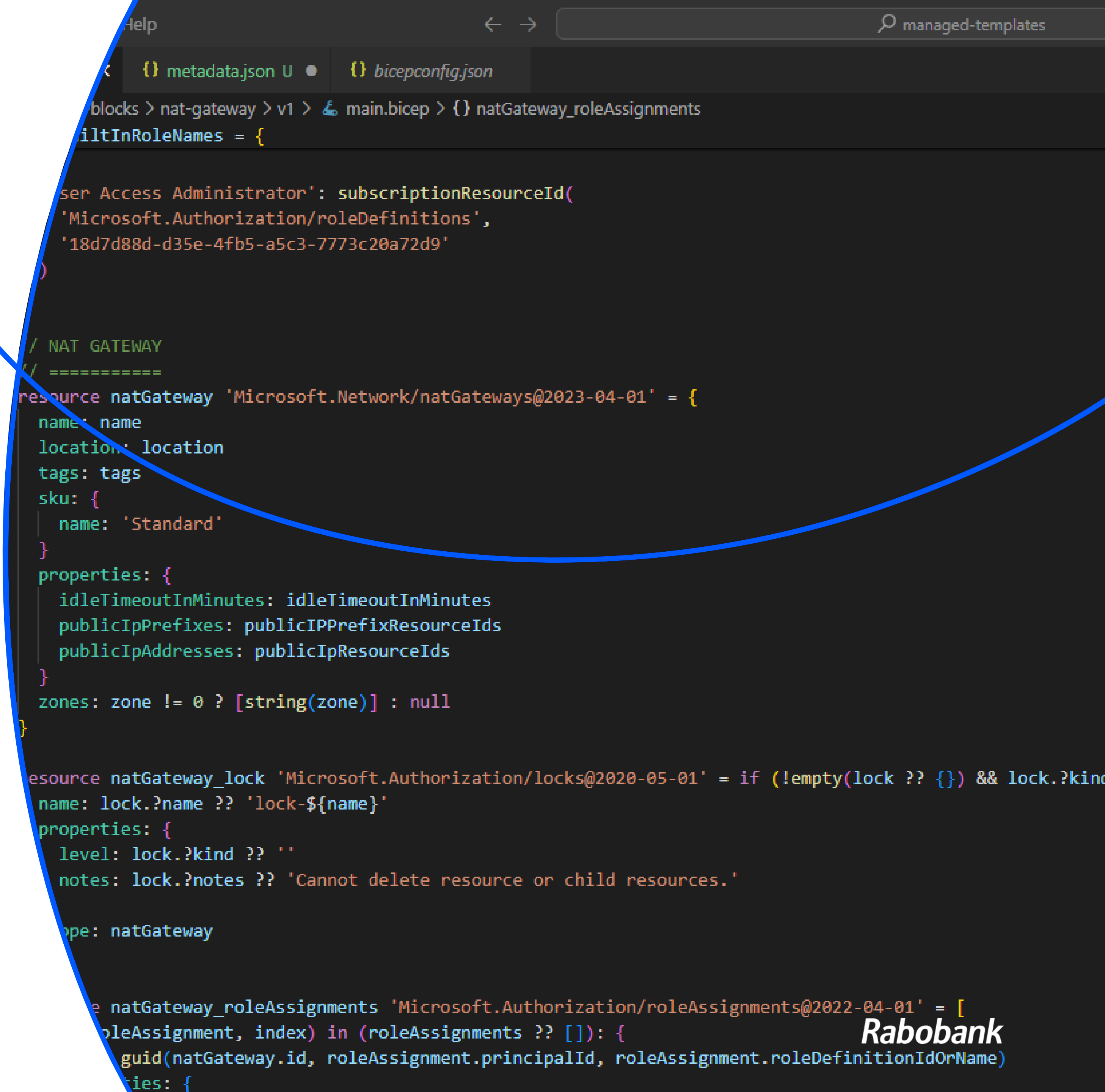
› Reference module invocation: (click to expand)

⌄ Example 1: Azure Front Door Quickstart (click to expand)

This parameter file has the required parameters for creating a simple Azure Front Door with it's child resources: origin group, origin, AFD endpoint, and security policy. This is a minimal example of a Azure Front Door resource. It only has a default non-Rabobank domain, and therefore doesn't need a certificate. Note that this example is not suitable for production use.

```
module exampleResource 'ts:                              3/building-blocks-v1/front-door-cdn:1.0.0' = {
    name: '<name of the deployment>'
    params: {
        name: 'myAfdProfileName'
        sku: 'Standard_AzureFrontDoor'
        originGroups: [
            {
                name: 'myOriginGroupName'
                loadBalancingSettings: {
                    additionalLatencyInMilliseconds: 0
                    sampleSize: 4
                    successfulSamplesRequired: 3
                }
                origins: [
                    {
```

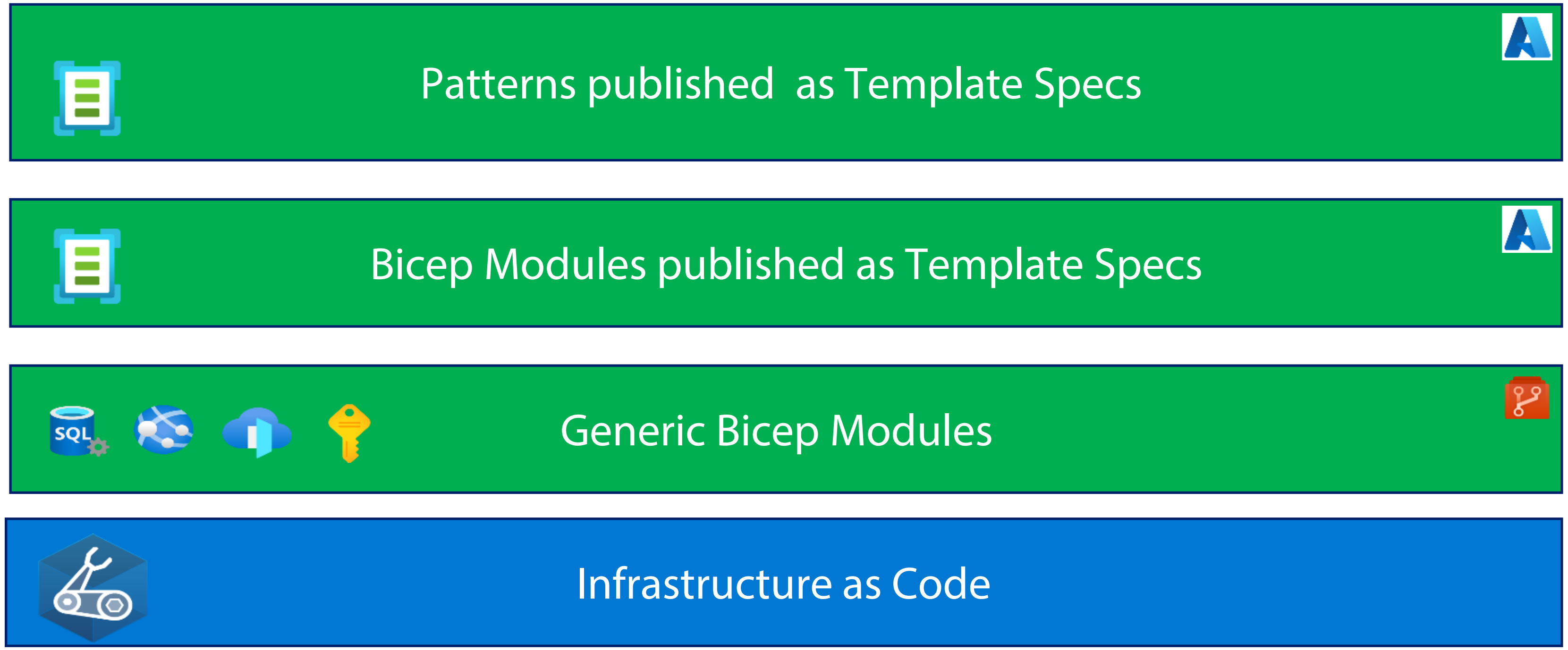
Bear with us… Live Demo!



Help
managed-templates
{} metadata.json U ●
{} bicepconfig.json
blocks > nat-gateway > v1 > main.bicep > {} natGateway_roleAssignments

```
iltInRoleNames = {

ser Access Administrator': subscriptionResourceId(
'Microsoft.Authorization/roleDefinitions',
'18d7d88d-d35e-4fb5-a5c3-7773c20a72d9'
)


/ NAT GATEWAY
// ===========
resource natGateway 'Microsoft.Network/natGateways@2023-04-01' = {
 name: name
 location: location
 tags: tags
 sku: {
   name: 'Standard'
 }
 properties: {
   idleTimeoutInMinutes: idleTimeoutInMinutes
   publicIpPrefixes: publicIPPrefixResourceIds
   publicIpAddresses: publicIpResourceIds
 }
 zones: zone != 0 ? [string(zone)] : null
}

esource natGateway_lock 'Microsoft.Authorization/locks@2020-05-01' = if (!empty(lock ?? {}) && lock.?kind
 name: lock.?name ?? 'lock-${name}'
 properties: {
   level: lock.?kind ?? ''
   notes: lock.?notes ?? 'Cannot delete resource or child resources.'

 ope: natGateway


e natGateway_roleAssignments 'Microsoft.Authorization/roleAssignments@2022-04-01' = [
oleAssignment, index) in (roleAssignments ?? []): {
 guid(natGateway.id, roleAssignment.principalId, roleAssignment.roleDefinitionIdOrName)
ies: {
```

Azure Template Specs



Patterns published as Template Specs

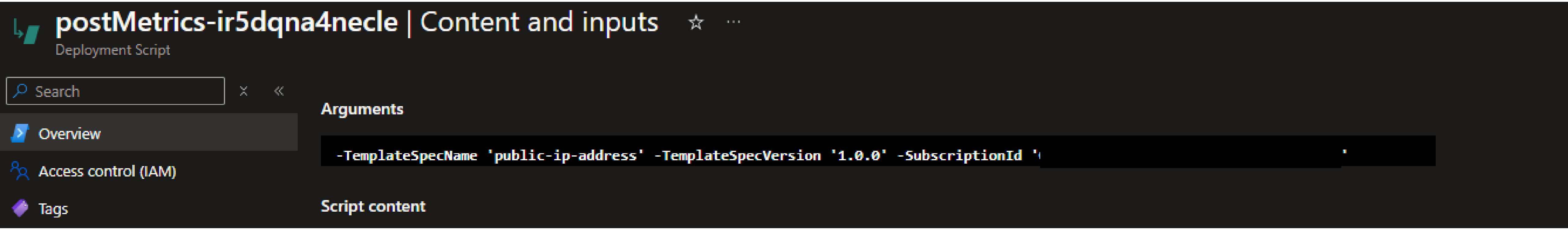
Bicep Modules published as Template Specs

Generic Bicep Modules

Infrastructure as Code

Understanding our consumers

What is consumed and by whom?



postMetrics-ir5dqna4necle | Content and inputs

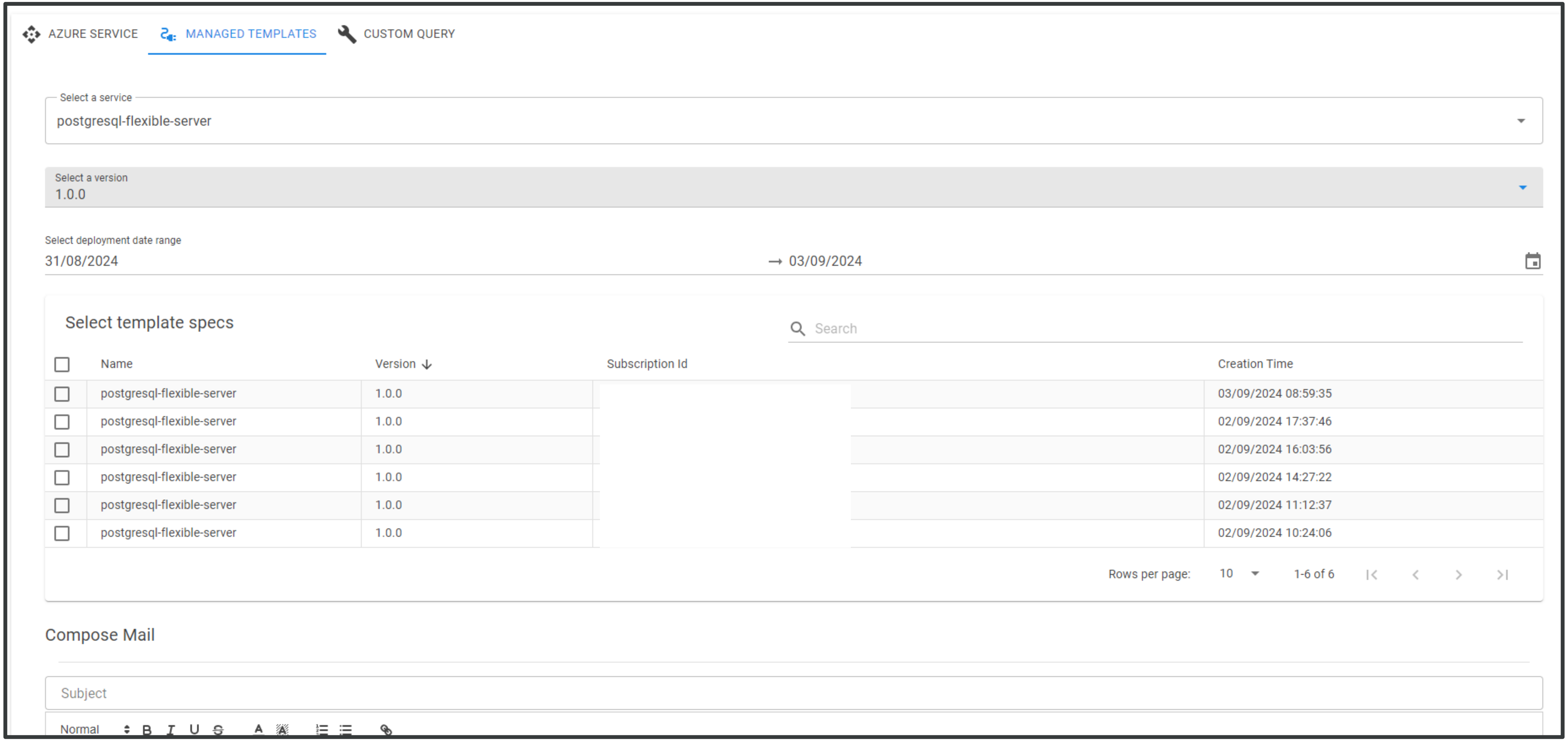
Deployment Script

Arguments

```
-TemplateSpecName 'public-ip-address' -TemplateSpecVersion '1.0.0' -SubscriptionId '[redacted]'
```

Script content

What is the data used for?



Select a service
postgresql-flexible-server

Select a version
1.0.0

Select deployment date range
31/08/2024 → 03/09/2024

Select template specs

□	Name	Version ↓	Subscription Id	Creation Time
□	postgresql-flexible-server	1.0.0	[redacted]	03/09/2024 08:59:35
□	postgresql-flexible-server	1.0.0	[redacted]	02/09/2024 17:37:46
□	postgresql-flexible-server	1.0.0	[redacted]	02/09/2024 16:03:56
□	postgresql-flexible-server	1.0.0	[redacted]	02/09/2024 14:27:22
□	postgresql-flexible-server	1.0.0	[redacted]	02/09/2024 11:12:37
□	postgresql-flexible-server	1.0.0	[redacted]	02/09/2024 10:24:06

Rows per page: 10 1-6 of 6

Compose Mail

Subject

Challenges

Balance between freedom and standardization

Shift in way of work

Example -> product

Not every resource type has an AVM module

Shift existing users

Guidance on Azure

Compliant Azure Service for teams to deploy

What about multiple resources?

Cognitive load?

Can this be less?

Thank you for your attention

Rabobank

Break

Drinks are outside of the room