Azure basic concepts. Deployments and IAC

10 2022

Deployments way in Azure?

- Azure ARM templates
- Bicep
- Terraform
- PowerShell
- CLI
- Other

ARM templates

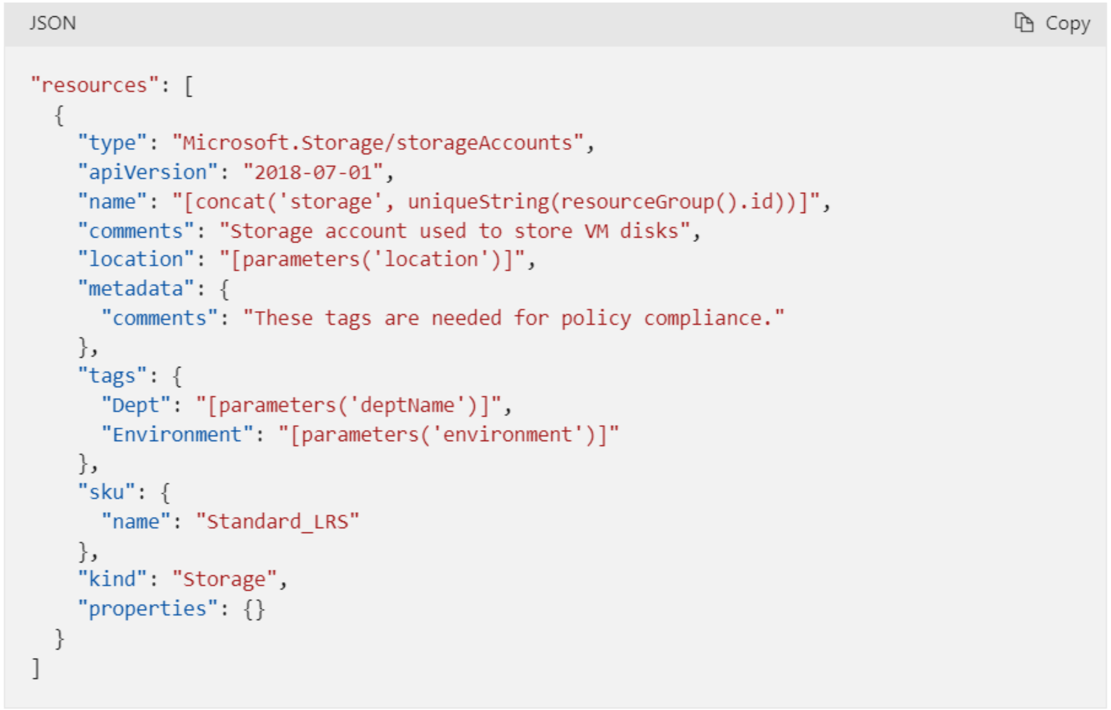
ARM templates advantages

- Can be downloaded before resource creation
- Can be downloaded after resource creation
- Nested templates
- No state file
- Can be integrated with terraform
- Support all resource types and api versions in Azure
- Official Azure support

ARM templates disadvantages

- Can be massive
- Sometimes arm templates hard to read
- Single cloud support only for Azure

ARM templates example



```json
"resources": [
  {
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2018-07-01",
    "name": "[concat('storage', uniqueString(resourceGroup().id))]",
    "comments": "Storage account used to store VM disks",
    "location": "[parameters('location')]",
    "metadata": {
      "comments": "These tags are needed for policy compliance."
    },
    "tags": {
      "Dept": "[parameters('deptName')]",
      "Environment": "[parameters('environment')]"
    },
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "properties": {}
  }
]
```

How to learn ARM?

- https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/template-tutorial-create-first-template?tabs=azure-powershell – official guides steps by steps for ARM templates
- https://azure.microsoft.com/en-us/resources/templates/ - ready templates for resources and different scenarios
- https://github.com/Azure/azure-quickstart-templates - quick start templates

Bicep

Bicep advantages

- Can be converted from ARM templates via decompile command
- Declarative and simple syntax
- Modularity
- Templates can be stored inside Azure Container Registry or Templates Specs
- No state file

Bicep disadvantages

- Can’t be generated from the Azure portal like arm templates
- Decompile option doesn’t fully convert from arm to bicep

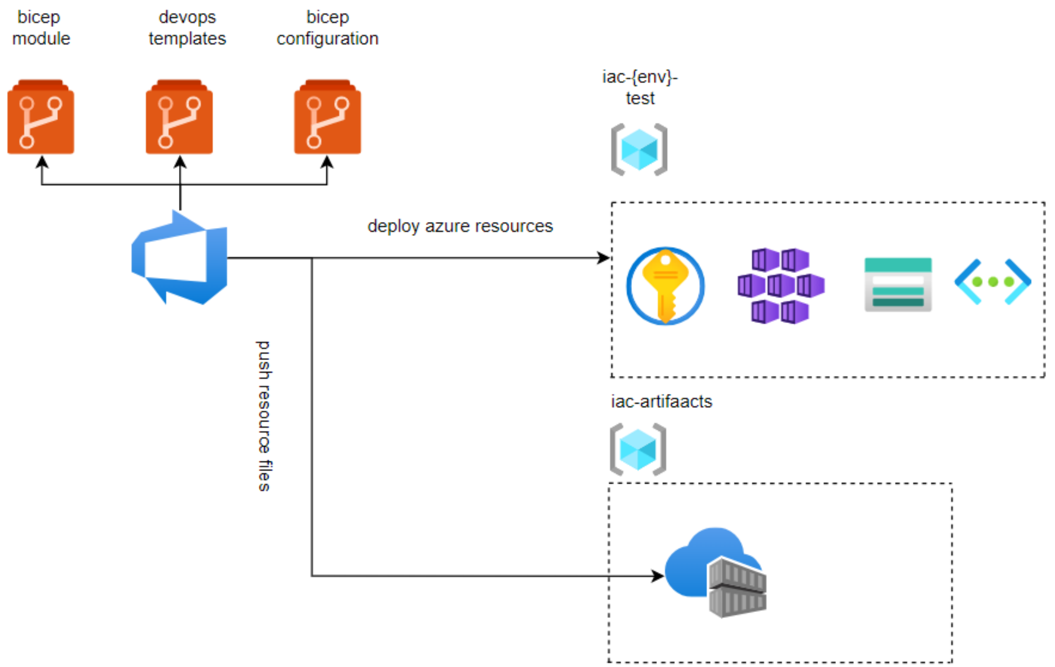
Bicep example

```
param location string = resourceGroup().location
param storageAccountName string = 'toylaunch${uniqueString(resourceGroup().id)}'

resource storageAccount 'Microsoft.Storage/storageAccounts@2021-06-01' = {
  name: storageAccountName
  location: location
  sku: {
    name: 'Standard_LRS'
  }
  kind: 'StorageV2'
  properties: {
    accessTier: 'Hot'
  }
}
```

How to learn Bicep?

- https://docs.microsoft.com/en-us/azure/azure-resource-manager/bicep/overview?tabs=bicep – Microsoft doc
- https://bicepdemo.z22.web.core.windows.net/ - examples
- https://github.com/Azure/azure-docs-bicep-samples - examples

Bicep example to be deployed



Terraform

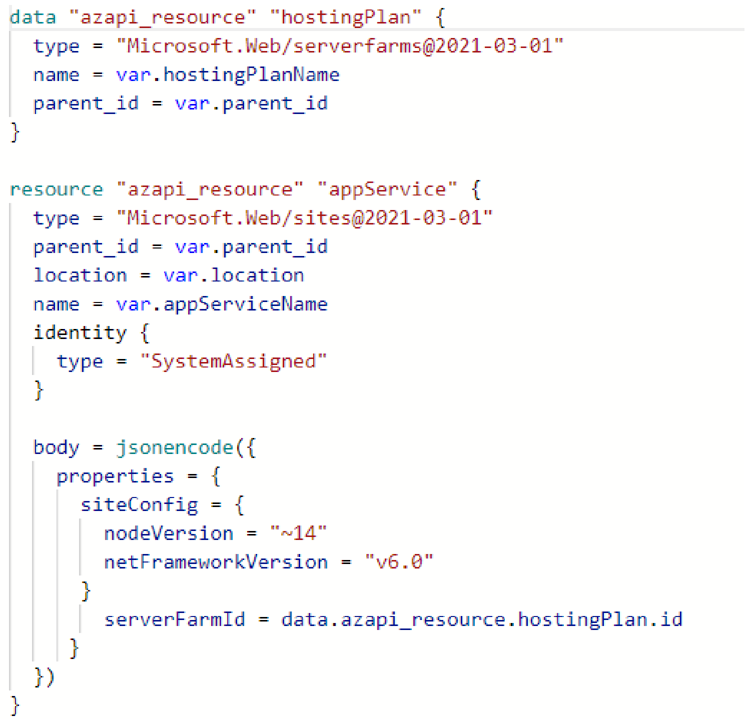
Terraform advantages

- Multi-cloud support
- Modularity and declarative syntax
- Terraform plan
- Multi provider deployment

Terraform disadvantages

- State file
- Needs to build secure place for storing state file
- Bugs
- Legacy stuff inside modules
- Not all apis is supported by azapi module

Terraform example

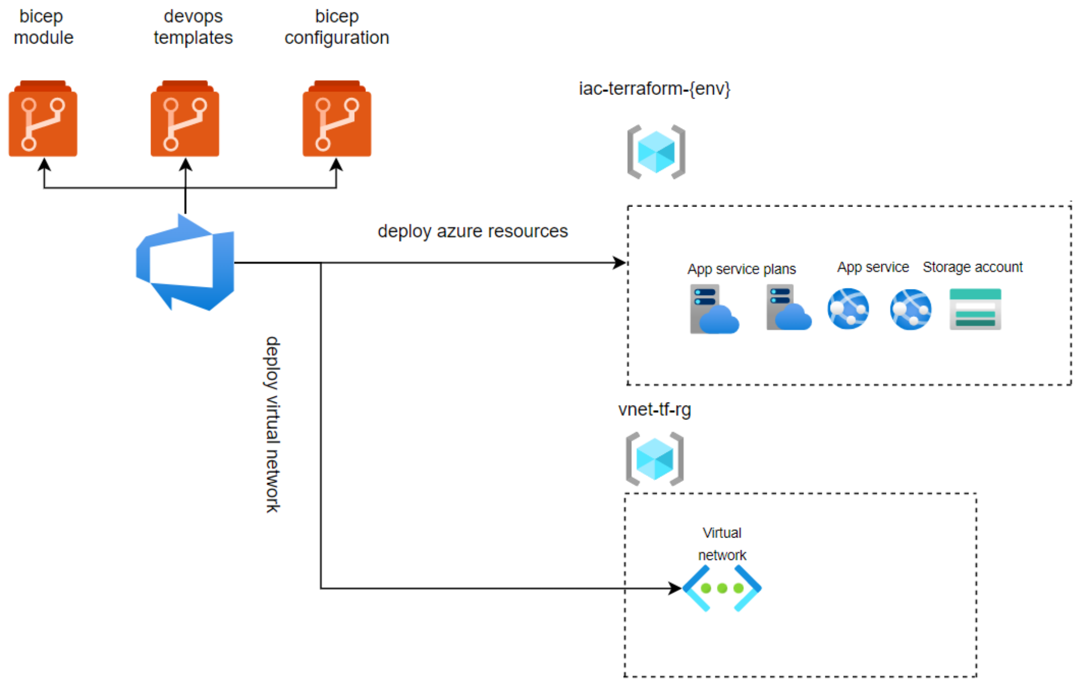


```
data "azapi_resource" "hostingPlan" {
  type = "Microsoft.Web/serverfarms@2021-03-01"
  name = var.hostingPlanName
  parent_id = var.parent_id
}

resource "azapi_resource" "appService" {
  type = "Microsoft.Web/sites@2021-03-01"
  parent_id = var.parent_id
  location = var.location
  name = var.appServiceName
  identity {
    type = "SystemAssigned"
  }

  body = jsonencode({
    properties = {
      siteConfig = {
        nodeVersion = "~14"
        netFrameworkVersion = "v6.0"
      }
        serverFarmId = data.azapi_resource.hostingPlan.id
      }
  })
}
```

How to learn Terraform?

- https://learn.hashicorp.com/collections/terraform/azure-get-started
- https://learn.microsoft.com/en-us/azure/developer/terraform/

Terraform example to be deployed



Questions