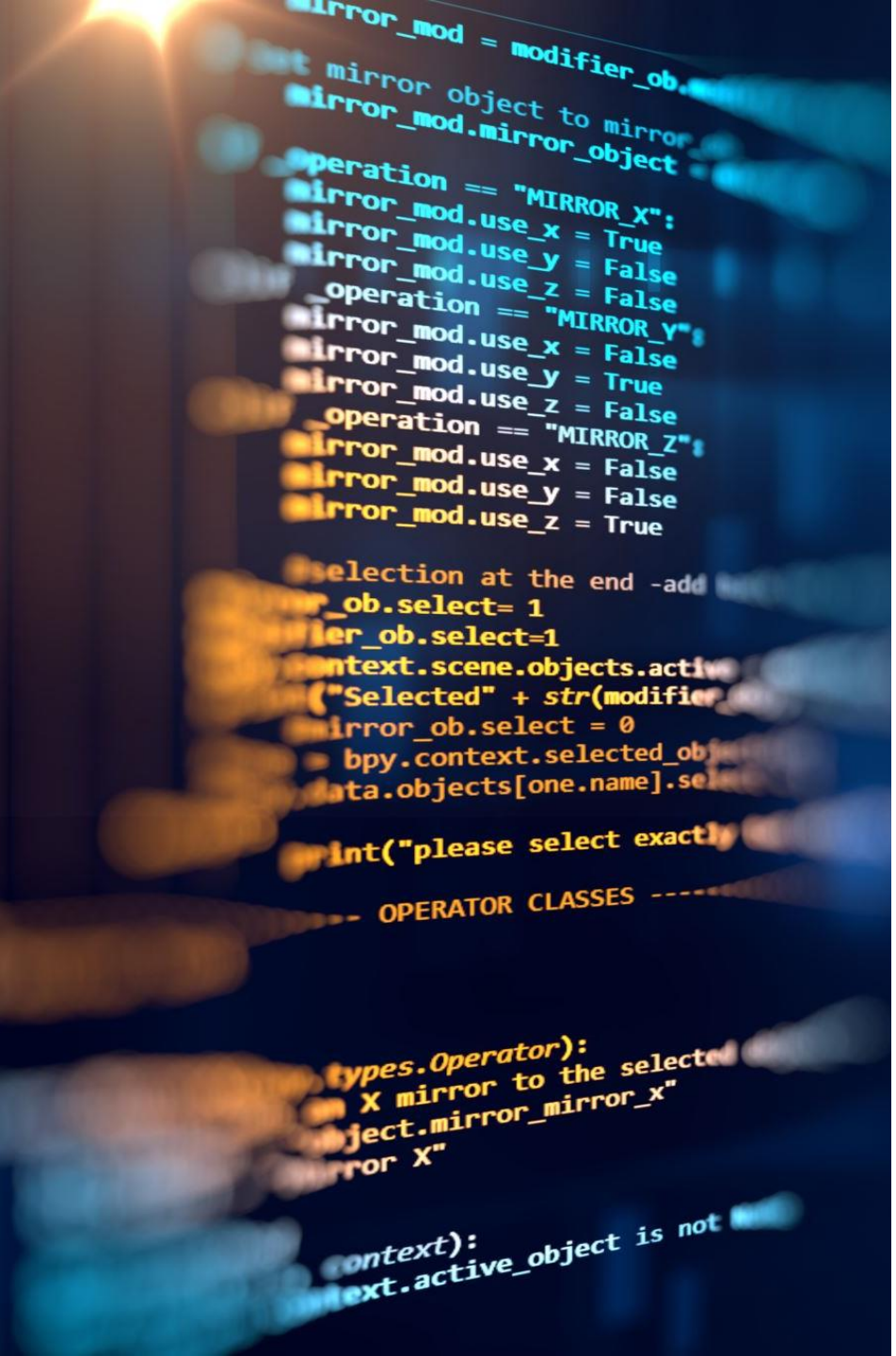
Bicep

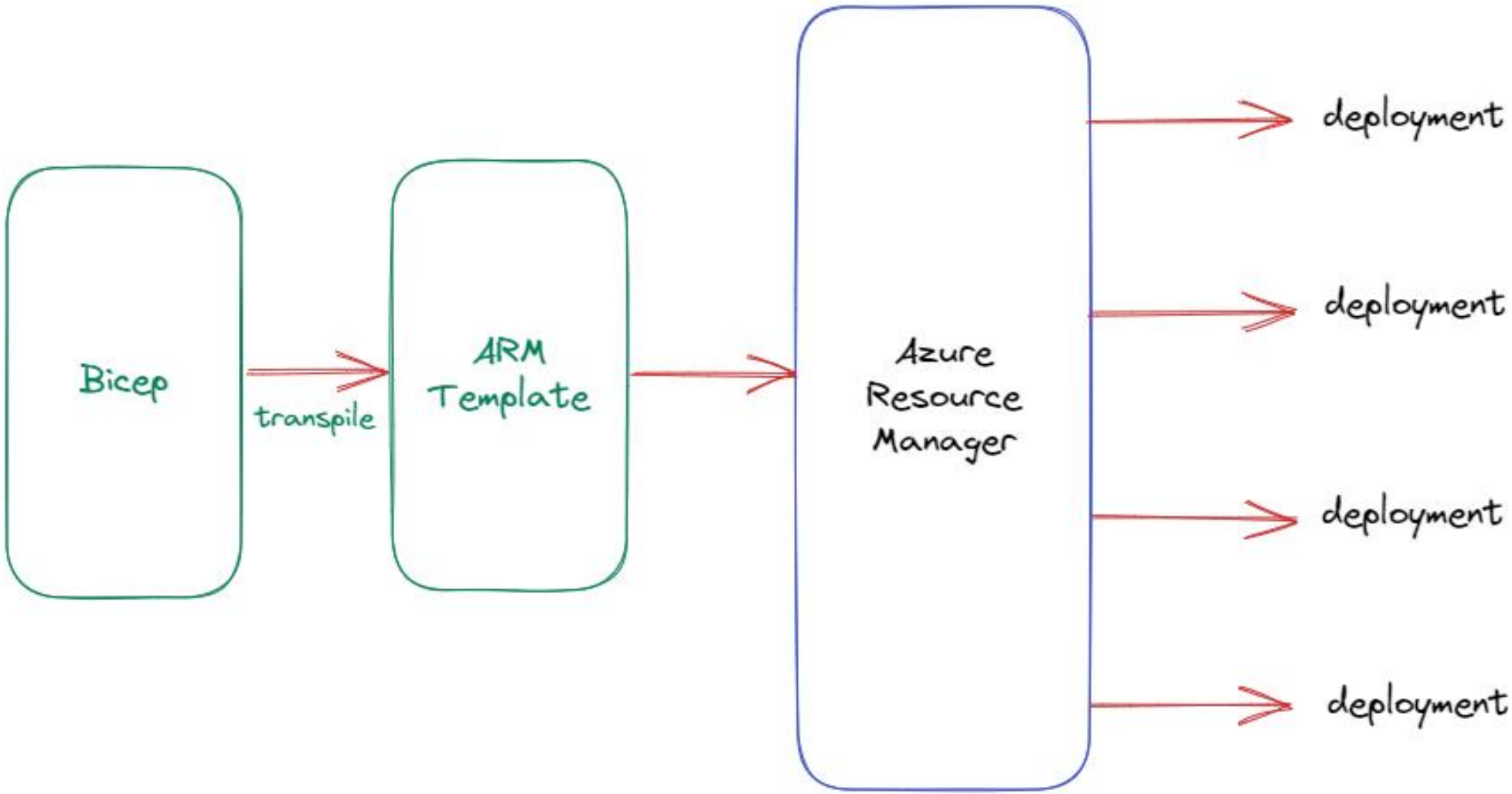
Getting Started



What is Bicep?

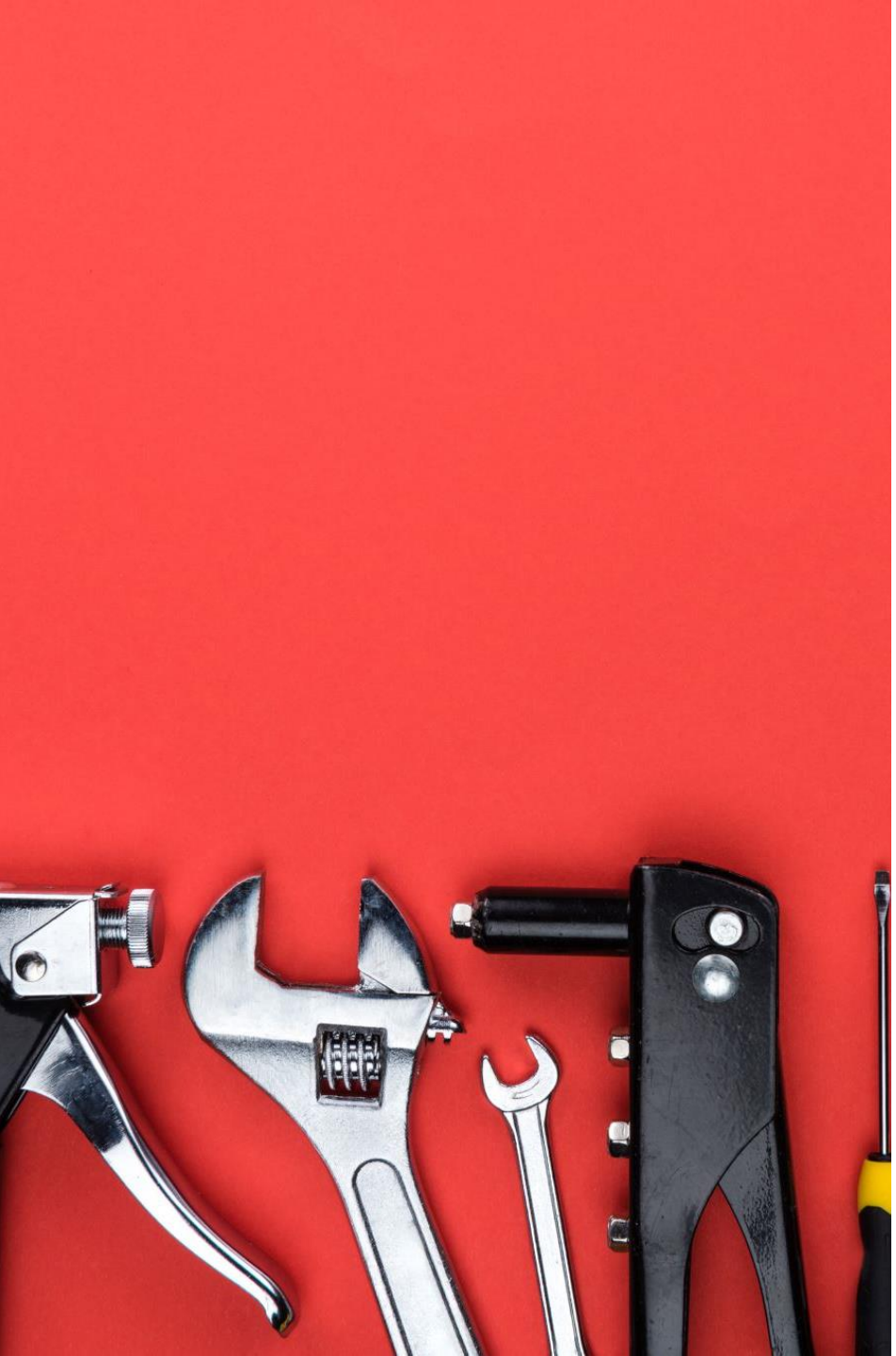
- Azure Resource Manager templates
  - JSON interpreted by ARM engine ☑
  - Authoring experience ✕

- Bicep = DSL (domain specific language)
  - Abstraction layer on top of ARM templates
  - Cleaner syntax, better type safety
  - Better support for modules


Bicep
transpile
ARM Template
Azure Resource Manager
deployment
deployment
deployment
deployment

```
{
  "type": "Microsoft.App/managedEnvironments",
  "apiVersion": "2022-03-01",
  "name": "[format('{0}-aca-env', parameters('parPrefix')
  "location": "[parameters('parLocation')]",
  "properties": {
    "appLogsConfiguration": {
      "destination": "log-analytics",
      "logAnalyticsConfiguration": {
        "customerId": "[reference(resourceId('Microsoft.O
        "sharedKey": "[listKeys(resourceId('Microsoft.Ope
      }
    },
    "daprAIConnectionString": "[reference(resourceId('Mic
  },
  "dependsOn": [
    "[resourceId('Microsoft.Insights/components', format(
    "[resourceId('Microsoft.OperationalInsights/workspace
  ]
},
{
  "type": "Microsoft.App/containerApps",
  "apiVersion": "2022-06-01-preview",
  "name": "[format('{0}-device-mgr', parameters('parPrefi
  "location": "[parameters('parLocation')]",
  "properties": {
    "managedEnvironmentId": "[resourceId('Microsoft.App/m
    "configuration": {
      "activeRevisionsMode": "Single",
      "dapr": {
        "enabled": true,
        "appPort": 5000,
```

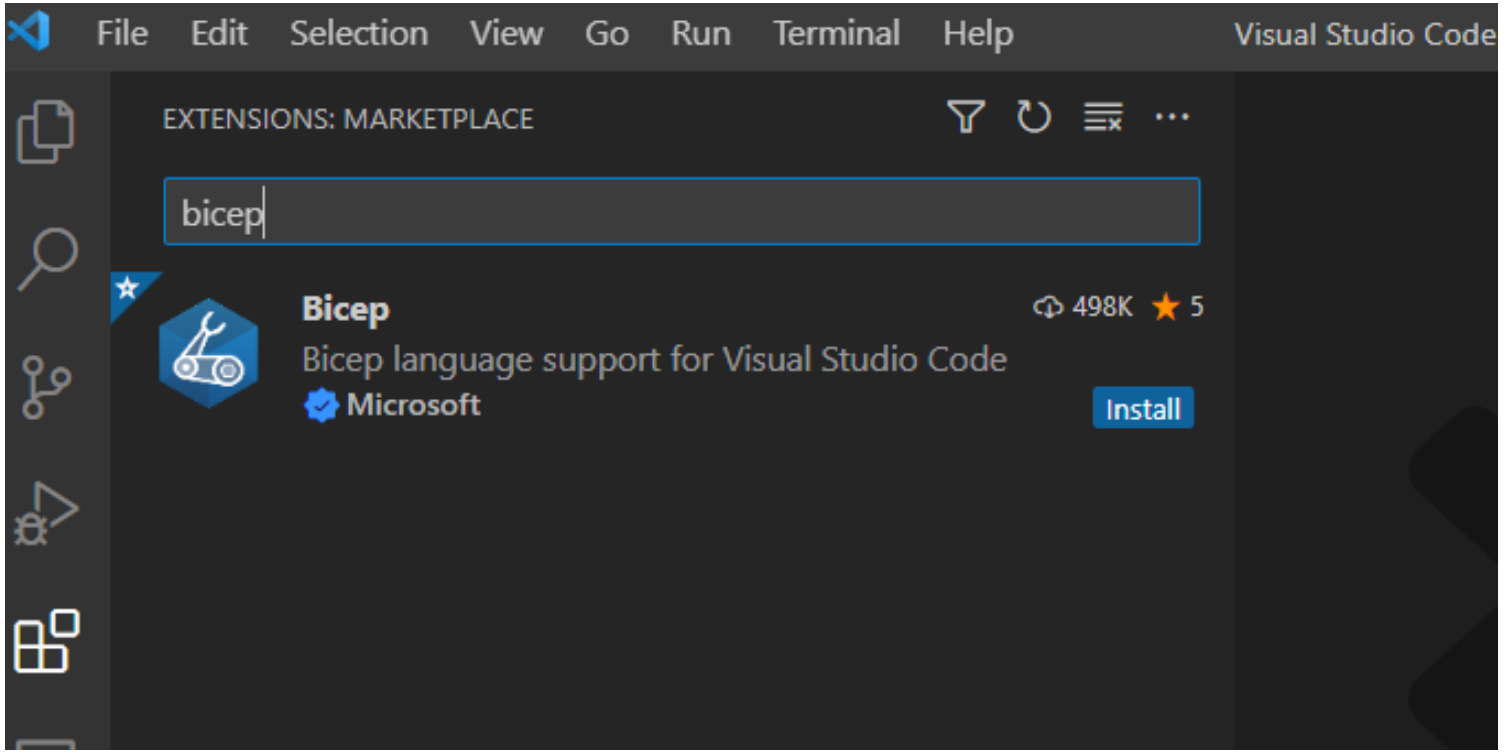
```
resource acaEnv 'Microsoft.App/managedEnvironments@2022-03-01'
  name: '${parPrefix}-aca-env'
  location: parLocation
  properties: {
    appLogsConfiguration: {
      destination: 'log-analytics'
      logAnalyticsConfiguration: {
        customerId: logAnalyticsWorkspace.properties.customerI
        sharedKey: listKeys(logAnalyticsWorkspace.id, logAnaly
      }
    }
    daprAIConnectionString: appInsights.properties.ConnectionS

  }
}

// deploy devicemgr as a container app
resource deviceMgr 'Microsoft.App/containerApps@2022-06-01-pre
  name: '${parPrefix}-device-mgr'
  location: parLocation
  properties: {
    managedEnvironmentId: acaEnv.id
    configuration: {
      activeRevisionsMode: 'Single'
      dapr: {
        enabled: true
        appPort:5000
        appId: 'devicemgr'
      }
      ingress: {
        external: false
        targetPort: 5000
```



What do you need?

- Install the Azure CLI (https://aka.ms/azcli)
- Either:
  - Run **az bicep install**
  - To update to a new version: **az bicep upgrade**
- Or:
  - Use the Azure Bicep CLI installer: https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/install#install-manually

VS Code Extension



Writing your first template: storage account

```
resource storageaccount 'Microsoft.Storage/storageAccounts@2022-09-01' = {
        name: 'name'
        location: 'westeurope'
        kind: 'StorageV2'
        sku: {
                name: 'Premium_LRS'
        }
}
```

Linter does not like this

See: 01

Adding a parameter

```
param parLocation string = 'westeurope'

resource storageaccount 'Microsoft.Storage/storageAccounts@2022-09-01' = {
          name: 'name'
          location: parLocation
          kind: 'StorageV2'
          sku: {
                    name: 'Premium_LRS'
          }
}
```

See 02

Deployment

```
RG=demo-RG
LOCATION=westeurope

# Create a resource group.
az group create --name $RG --location $LOCATION

# Deploy bicep
az deployment group create --resource-group $RG --template-file
02/main.bicep
```

Deployment at resource group level

Deployment Scopes

- Set with: `targetScope = '<scope>'`
- Possible scopes:
  - tenant
  - managementGroup
  - subscription
  - resourceGroup (default)

See 02a

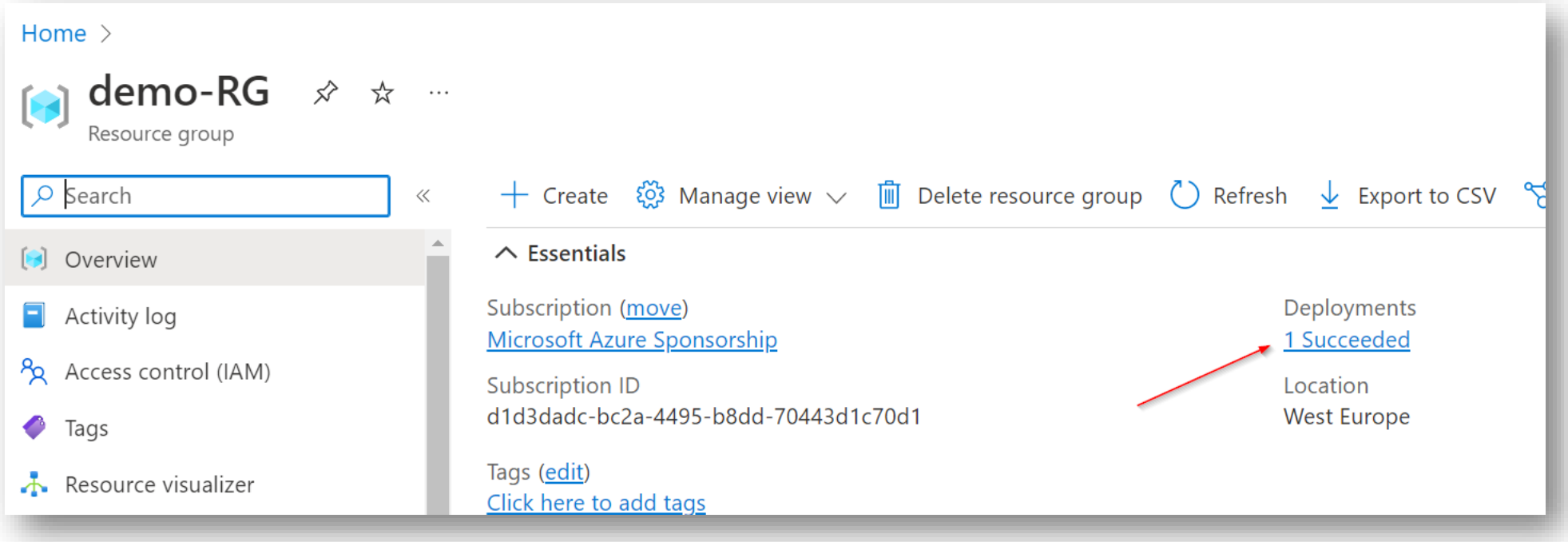
Setting a parameter

To set the parameter parLocation during deployment, we can use:

```
LOCATION=westeurope
az deployment group create --resource-group $RG --template-file 02/main.bicep \
        --parameters parLocation=$LOCATION
```

Checking the Deployment

- Check the JSON output
- Check the Azure Portal



Tips

- Run **bicep build <.bicep file>** to convert to ARM
  - The JSON result can also be deployed with **az deployment group**
- Run **bicep decompile** to decompile ARM into Bicep
  - Best effort process
- Use **Insert Resource** in VS Code to create a Bicep resource from a resource in Azure
  - Requires the resource ID

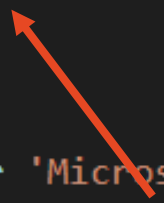
Creating a storage container in a storage account

```
resource container
'Microsoft.Storage/storageAccounts/blobServices/containers@2021-04-01' = {
        name: '${storageaccount.name}/default/mycontainer'
        properties: {
        publicAccess: 'None'
        }
}
```

⚠ See 03 and 04 in sample repository

Dependencies

```
param parLocation string = 'westeurope'

resource storageaccount 'Microsoft.Storage/storageAccounts@2022-09-01' = {
  name: 'stggeba266372'
  location: parLocation
  kind: 'StorageV2'
  sku: {
    name: 'Premium_LRS'
  }
}

resource container 'Microsoft.Storage/storageAccounts/blobServices/containers@2021-04-01' = {
  name: '${storageaccount.name}/default/mycontainer'
  properties: {
    publicAccess: 'None'
  }
}
```



Conditional Deployment

Use **if** in a **resource** declaration:

```
resource kv 'Microsoft.KeyVault/vaults@2022-07-01' = if (parDeployKeyVault) {
  name: 'kvgeba266372'
  location: parLocation
  dependsOn: [
    storageaccount
  ]
  properties: {
    sku: {
      family: 'A'
      name: 'standard'
    }
    tenantId: subscription().tenantId
    enableRbacAuthorization: true
  }

}
```

Parameters file

- Specify parameters in a json file
- You can combine a parameters file with individual command line parameters

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "parLocation": {
      "value": "westeurope"
    },
    "parDeployKeyVault": {
      "value": true
    }
  }
}
```

⚠ TIP: generate from VS Code

Parameters and decorators

- Parameter types: string, int, bool, array, object
- Decorators:
  - @description()
  - @allowed([])
  - @secure()
  - @minValue()
  - @maxValue()

See 07-params

Deploy multiple resources

Use **for** with a resource

```
// create multiple storage accounts based on the array
resource stgs 'Microsoft.Storage/storageAccounts@2022-09-01' = [for stg in parStorageAccounts: {
  name: stg.name
  kind: 'StorageV2'
  location: parLocation
  sku: {
    name: stg.sku
```

See 08-multiple

Using modules

- Modules are bicep templates that can be used by other templates
- Bicep code in a module is not different from a directly executed template
- How to call the module? (See 09-modules)

```
module sa 'storage.bicep' = {
  name: 'sa'
  params: {
    parStorageAccounts: parStorageAccounts
    parLocation: parLocation
  }
}
```

Module registry vs local module

- Share modules within the organization with a **private module registry**
- Requires **Azure Container Registry** (ACR)
  - You need role to push images
- Command to use:

  az bicep publish --file storage.bicep --target br:regname.azurecr.io/bicep/modules/storage:v1

- Refer to module:

  module name 'br:regname.azurecr.io/bicep/modules/storage:v1' = { ... }

Bicep Public Registry

- See https://github.com/Azure/bicep-registry-modules

- Example usage:

```
module myenv 'br/public:app/dapr-containerapps-environment:1.0.1' = {
  name: 'state'
  params: {
    location: location
    nameseed: 'stateSt1'
    applicationEntityName: 'appdata'
    daprComponentType: 'state.azure.blobstorage'
    daprComponentScopes: [
      'nodeapp'
    ]
  }
}
```

What-If Analysis

```
RG=demo-RG
LOCATION=westeurope

# Create a resource group.
az group create --name $RG --location $LOCATION

# Bicep what if analysis
az deployment group what-if --resource-group $RG --template-file
./main.bicep \
        --parameters parLocation=$LOCATION
```

Creating unique strings

- Creates a deterministic hash string based on the values provided as parameters
- For example:
  - uniqueString(resourceGroup().id)
  - uniqueString(subscription().subscriptionId)
- Ensures that redeployment does not create new resources

See 11-uniqueString

Examining the VM deployment

- Virtual network and subnet
- Network Security Group
- Availability Set
- Virtual Machines
- Load Balancer
- Storage Account
- Front Door