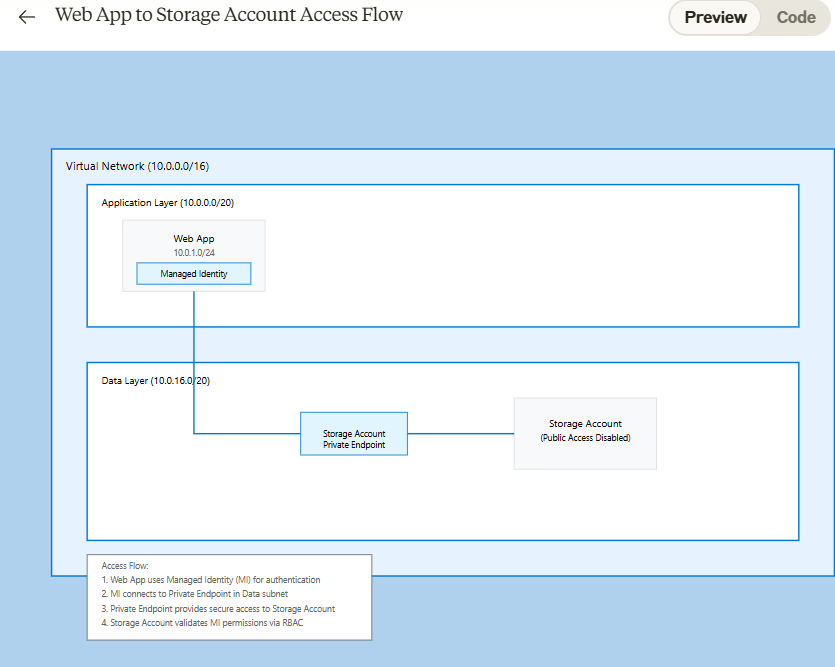
**How WebApp access storage using Managed Identity:**

****

1. First, create Storage Account with private endpoint:

// Storage Account with no public access

resource storageAccount 'Microsoft.Storage/storageAccounts@2023-01-01' = {

name: storageAccountName

properties: {

publicNetworkAccess: 'Disabled' // Disable public access

networkAcls: {

defaultAction: 'Deny' // Deny all public traffic

}

}

}

// Private Endpoint for Storage in the Data subnet

resource **privateEndpoint** 'Microsoft.Network/privateEndpoints@2023-05-01' = {

name: 'pe-storage'

location: location

properties: {

subnet: {

id: **dataSubnetId** // This is in the data layer subnet (10.0.16.0/20)

}

privateLinkServiceConnections: [

{

name: 'storage-connection'

properties: {

privateLinkServiceId: storageAccount.id

groupIds: [

'blob'

]

}

}

]

}

}

1. Enable Managed Identity in WebApp

resource webApp 'Microsoft.Web/sites@2022-09-01' = {

name: webAppName

identity: {

**type: 'SystemAssigned'** // Enable managed identity

}

properties: {

virtualNetworkSubnetId: webAppSubnetId // This is in application subnet (10.0.1.0/24)

vnetRouteAllEnabled: true // Route all traffic through VNet

}

}

1. Then in Storage access the WebApp managed identity and assign role of Storage blob data contributor role.

// Give Web App's managed identity access to Storage

resource storageRoleAssignment 'Microsoft.Authorization/roleAssignments@2022-04-01' = {

name: guid(webApp.id, storageAccount.id, 'StorageBlobDataContributor')

scope: storageAccount

properties: {

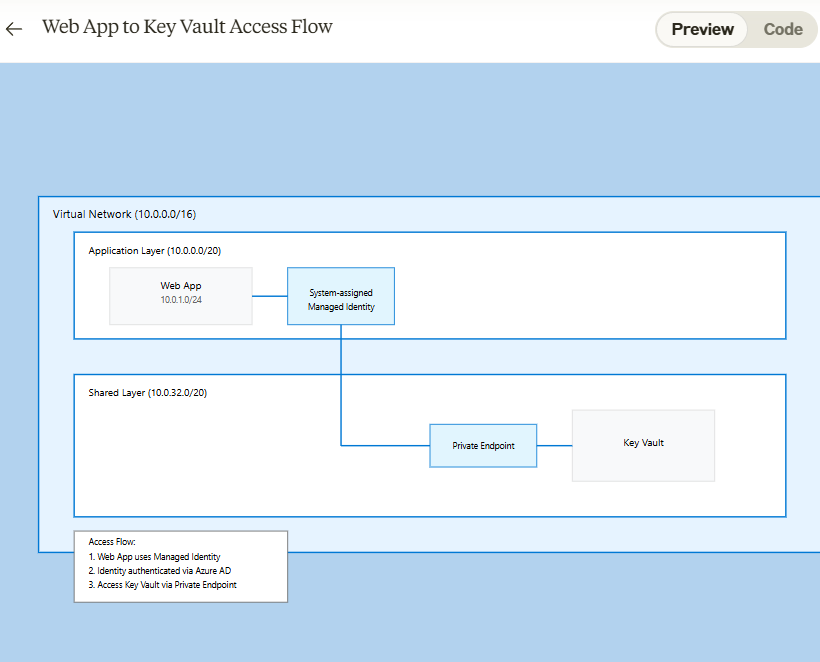
**principalId: webApp.identity.principalId**

**roleDefinitionId: 'ba92f5b4-2d11-453d-a403-e96b0029c9fe'** // Storage Blob Data Contributor

}

}

**How WebApp access the KeyVault**

****

1. First, create KeyVault with private endpoint in Shared Subnet

resource keyVault 'Microsoft.KeyVault/vaults@2023-02-01' = {

name: 'kv-${baseName}-${environment}'

properties: {

enableRbacAuthorization: true

publicNetworkAccess: 'Disabled'

networkAcls: {

defaultAction: 'Deny'

bypass: 'AzureServices'

}

}

}

resource privateEndpoint 'Microsoft.Network/privateEndpoints@2023-05-01' = {

name: 'pe-${keyVault.name}'

location: location

properties: {

subnet: {

**id: sharedSubnetId** // Subnet in shared layer

}

privateLinkServiceConnections: [

{

name: 'pe-${keyVault.name}'

properties: {

privateLinkServiceId: keyVault.id

groupIds: ['vault']

}

}

]

}

}

1. Enable Managed Identity in WebApp

resource webApp 'Microsoft.Web/sites@2022-09-01' = {

name: 'app-${baseName}-${environment}'

identity: {

**type: 'SystemAssigned'** // Enable managed identity

}

properties: {

// ... other properties

}

}

1. Grant Web App's managed identity access to Key Vault:

resource keyVaultRoleAssignment 'Microsoft.Authorization/roleAssignments@2022-04-01' = {

name: guid(keyVault.id, webApp.id, 'KeyVaultSecretsUser')

scope: keyVault

properties: {

**roleDefinitionId: subscriptionResourceId**('Microsoft.Authorization/roleDefinitions', '4633458b-17de-408a-b874-0445c86b69e6') // Key Vault Secrets User

**principalId: webApp.identity.principalId**

principalType: 'ServicePrincipal'

}

}

