Secure Azure AI Services

Saturday, June 22, 2024 4:38 PM

Consider authentication
By default Azure AI services resources is restricted by using subscription keys. Management of access to these keys is a primary consideration for security.

Regenerate Keys

You should regenerate keys regularly to protect against the risk of keys being shared with or accessed by unauthorized users. You can regenerate keys using the Azure portal, or using the `az cognitiveservices account keys regenerate` Azure command-line interface (CLI) command.
Each AI service is provided with two keys, enabling you to regenerate keys without service interruption. To accomplish this:

1. If you're using both keys in production, change your code so that only one key is in use. For example, configure all production applications to use key 1.
2. Regenerate key 2.
3. Switch all production applications to use the newly regenerated key 2.
4. Regenerate key 1
5. Finally, update your production code to use the new key 1.

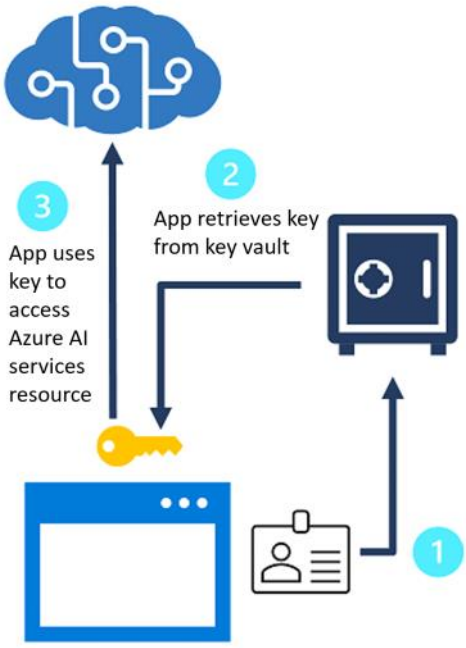
For example, to regenerate keys in the Azure portal, you can do the following:

1. In the Azure portal, go to your resource's Keys and Endpoint pane.
2. Then select Regenerate Key1 or select Regenerate Key2, depending on which one you want to regenerate at the time.

From <https://learn.microsoft.com/en-us/training/modules/secure-ai-services/2-authentication>

Protect keys with Azure Key Vault:

Azure Key Vault is a secure service for storing secrets like passwords and keys in Azure. Access is controlled via security principals authenticated with Microsoft Entra ID. Administrators can assign a managed identity to applications, allowing them to access specific secrets in the key vault without hard-coding them in applications or configuration files. This reduces the risk of unauthorized access. For instance, AI service subscription keys can be stored in Azure Key Vault, and client applications can use managed identities to retrieve these keys securely.



Token-based authentication

When using the REST interface, some AI services support (or even *require*) token-based authentication. In these cases, the subscription key is presented in an initial request to obtain an authentication token, which has a valid period of 10 minutes. Subsequent requests must present the token to validate that the caller has been authenticated.

When using an SDK, the calls to obtain and present a token are handled for you by the SDK.

From <https://learn.microsoft.com/en-us/training/modules/secure-ai-services/2-authentication>

Microsoft Entra ID authentication

Azure AI services supports Microsoft Entra ID authentication, enabling you to grant access to specific service principals or managed identities for apps and services running in Azure.

Authenticate using service principals

The overall process to authenticate against Azure AI services using service principals is as follows:

Create a custom subdomain

Create a subdomain using the following code:
Set-AzContext -SubscriptionName <Your-Subscription-Name>

Then, you create your Azure AI services resources specifying a custom subdomain by running the following code:

$account = New-AzCognitiveServicesAccount -ResourceGroupName <your-resource-group-name> -name <your-account-name> -Type <your-account-type> -SkuName <your-sku-type> -Location <your-region> -CustomSubdomainName <your-unique-subdomain-name>

Once created, your subdomain name will be returned in the response

Assign a role to a service principal
You've created an Azure AI resource that is linked with a custom subdomain. Next, you assign a role to a service principal.

To start, you'll need to register an application. To do this, you run the following command:

```
$SecureStringPassword = ConvertTo-SecureString -String <your-password> -AsPlainText -Force

$app = New-AzureADApplication -DisplayName <your-app-display-name> -IdentifierUris <your-app-uris> -PasswordCredentials $SecureStringPassword
```

This creates the application resource.

Then you use the New-AzADServicePrincipal command to create a service principal and provide your application's ID:

```
New-AzADServicePrincipal -ApplicationId <app-id>
```

Finally, you assign the Cognitive Services Users role to your service principal by running:
```
New-AzRoleAssignment -ObjectId <your-service-principal-object-id> -Scope <account-id> -RoleDefinitionName "Cognitive Services User"
```

Authenticated using managed identities
Managed identities come in two types:

- System-assigned managed identity: A managed identity is created and linked to a specific resource, such as a virtual machine that needs to access Azure AI services. When the resource is deleted, the identity is deleted as well.
- User-assigned managed identity: The managed identity is created to be useable by multiple resources instead of being tied to one. It exists independently of any single resource.

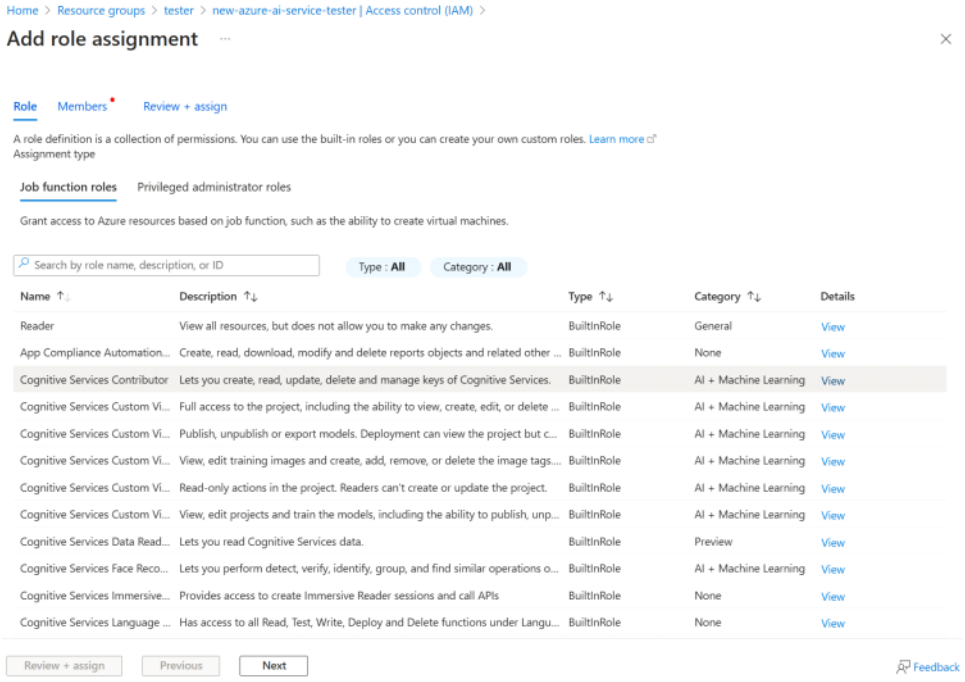
To enable a system-assigned identity for a VM to use Azure AI Services you need to do the following:

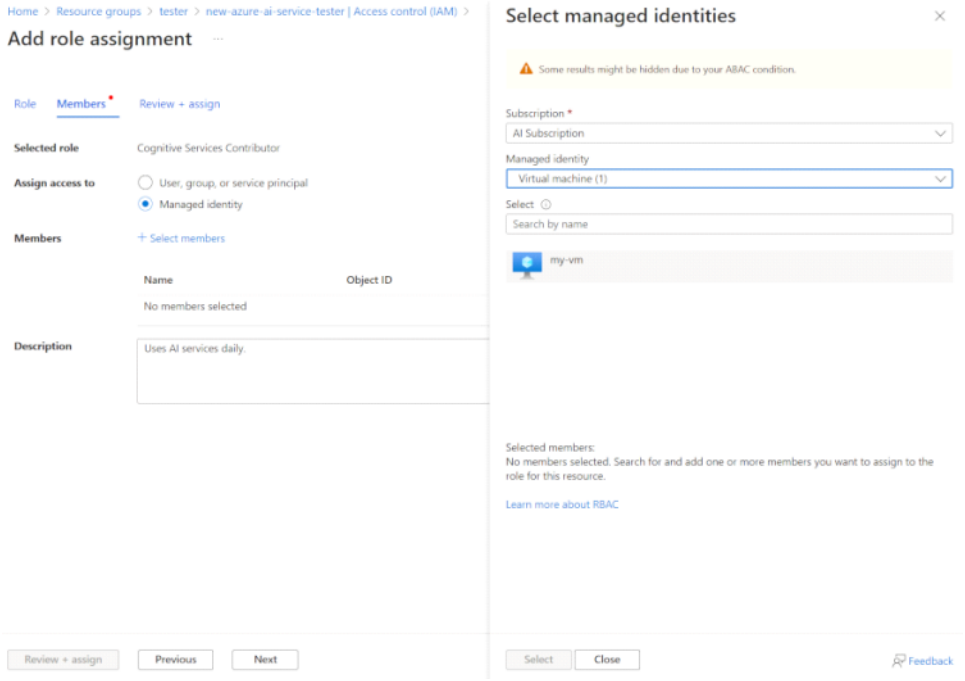
First make sure the Azure account has the vm contributor role. Then you can run the following command using Azure CLI :
```
az vm identity assign -g <my-resource-group> -n <my-vm>
```

Then you can grant access to Azure AI services in the Azure portal using the following:

1. Go to the Azure AI services resource you want to grant the virtual machine's managed identity access.
2. In the overview panel, select Access control (IAM).
3. Select Add, and then select Add role assignment.
4. In the Role tab, select Cognitive Services Contributor.
5. In the Members tab, for the Assign access to, select Managed identity. Then, select + Select members.
6. Ensure that your subscription is selected in the Subscription dropdown. And for Managed identity, select Virtual machine.

7. Select your virtual machine in the list, and select Select.
8. Finally, select Review + assign to review, and then Review + assign again to finish.





Implement network security

Network security is an important measure to ensure unauthorized users can't reach the services that you are protecting. Limiting what users can see is always a great idea, since they can’t compromise what they can’t see.

Apply network access restrictions

By default, Azure AI services are accessible from all networks. Some individual AI services resources (such as Azure AI Face service, Azure AI Vision, and others) can be configured to restrict access to specific network addresses - either public Internet addresses or addresses on virtual networks. With network restrictions enabled, a client trying to connect from an IP address that isn't allowed will receive an Access Denied error.