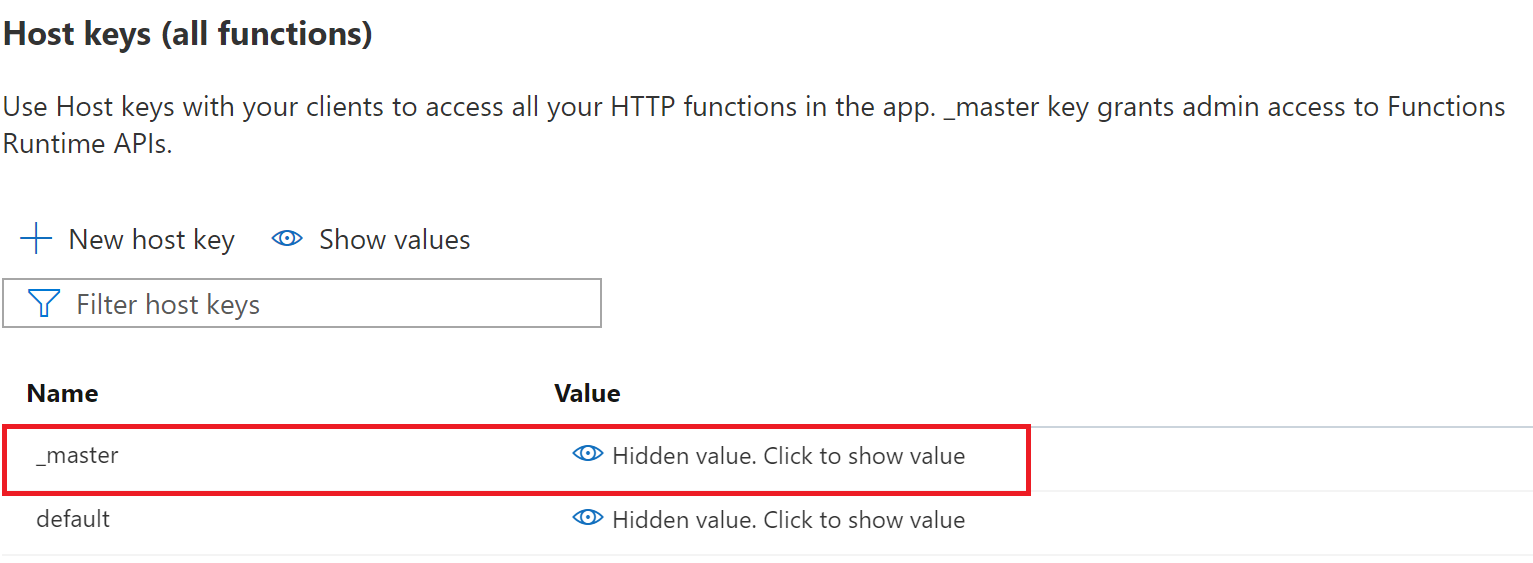
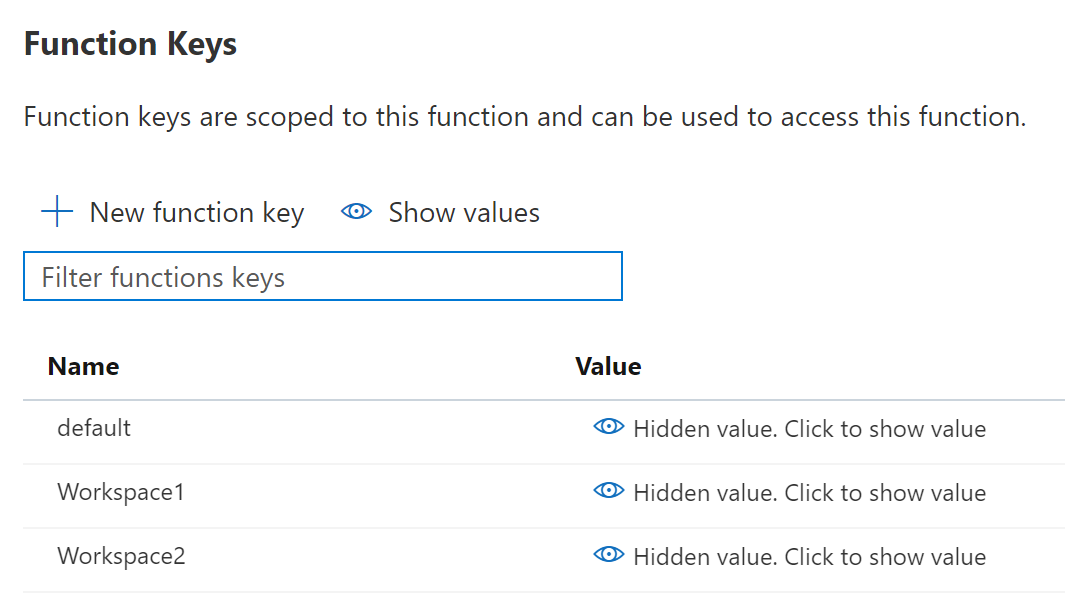
**Function Access Key Samples Pre-requisites**

1. Create an Azure Function App
2. Create a system-managed identity for the function app created in step 1
3. Create an Azure Key Vault
4. Give the system-managed identity of the function app permissions to read secrets from the Key Vault
5. Create a secret named AccessKeyManagementKey in the Key Vault and set its value to the \_master host key of the Azure Function



1. Modify the configuration settings in the local.settings.json file (or in the Function Configuration blade) and deploy your Azure Function to the function app created in step 1
2. Create a function key for the GetTokenCallAPI function for each workspace. The name of the function key should be the workspace name.



1. Register an application in Azure AD for each workspace. This will give each workspace a unique set of credentials to retrieve an access token from Azure AD. This will serve as the “OAuthClient” from the Snowflake examples. Make sure to create a client secret when you create the app registration and give it the appropriate permissions required to call your API.
2. For each workspace create a secret in the Key Vault named {workspacename}Secrets. The secret value should contain the client ID and client secret of the workspace’s Azure AD app registration as follows:

{"clientId":"{your client ID}","clientSecret":"your client secret}"}

1. Run the following command to obtain the list of possible outbound IP addresses that will be used by the Azure Function App. Save these as they will need to be added to the Function App’s Access Restrictions list to allow the function app to call itself to get a list of its function keys.

az webapp show --resource-group <group\_name> --name <app\_name> --query possibleOutboundIpAddresses --output tsv

1. Set the Access Restrictions on the Azure Function App to include the IP addresses of any ML workspaces, VMs, etc. that will need to call the function. Make sure to include the IP addresses from step 10.

