atingupta2005 Update README.md 3 hours ago

109 lines (89 loc) · 3.42 KB

Preview Code Blame Raw

How to use RBAC in Cosmos DB

References:

- https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-setup-rbac#role-assignments
- https://learn.microsoft.com/en-us/cli/azure/cosmosdb/sql/role/assignment?view=azure-cli-latest#az-cosmosdb-sql-role-assignment-create
- https://cosmos.azure.com/?feature.enableAadDataPlane=true

Create a new service principal if required in AAD

SP Details:

```
servicePrincipal: 6bb2f9af-a0af-4c32-a5ec-5f7011d37551
Tanent ID: 7aae85f5-b903-4220-a8c6-678dc4b73a3f
Secret: b0.8Q~AL_3pqoPLAqWMxR7MW28GQLWYaP6cVoaEZ
```

Open Azure CLI and run below commands

```
subscription_id="f22f6f07-9e0b-47e3-bc12-f333f88d9d70"
az account set --subscription "$subscription_id"
```

Create a file named myrole.json and put below content in it

```
{
    "RoleName": "MyReadOnlyRole",
    "Type": "CustomRole",
    "AssignableScopes": ["/"],
    "Permissions": [{
```

```
        "DataActions": [
            "Microsoft.DocumentDB/databaseAccounts/readMetadata",
            "Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/items/read",

"Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/executeQuery",

"Microsoft.DocumentDB/databaseAccounts/sqlDatabases/containers/readChangeFeed"
        ]
    }]
}
```

Set environment variables

```
resourceGroupName='shravya'
accountName='kirandb'
```

Create the custom role

```
az cosmosdb sql role definition create --account-name $accountName --resource-group
$resourceGroupName --body @myrole.json
```

```
az cosmosdb sql role definition list -a $accountName -g $resourceGroupName
```

Save Role ID in below variable

```
readOnlyRoleDefinitionId="00000000-0000-0000-0000-000000000001"
```

Show the details of the role created

```
role_id="00000000-0000-0000-0000-000000000001"
az cosmosdb sql role definition show --account-name $accountName --resource-group
$resourceGroupName --id $role_id
```

Set environment variable with Service Principal

```
principalId='56df8665-eaeb-41ff-9b9e-a39f35e9c07a'
```

Delete (If any) and then Assign role to Cosmos DB

```
az cosmosdb sql role assignment delete --account-name $accountName --resource-group
$resourceGroupName --role-assignment-id $readOnlyRoleDefinitionId
```

```
az cosmosdb sql role assignment create --account-name $accountName --resource-group
$resourceGroupName --scope "/" --role-definition-id $readOnlyRoleDefinitionId --
```

```
principal-id $principalId
```

List roles to confirm assignment

```
az cosmosdb sql role assignment list --account-name $accountName --resource-group
$resourceGroupName
```

Set environment variable with role assignment id from output of previous command.

```
role_assignment_id="81be1ab5-498b-468b-af2a-77e3879cdb86"
```

Check if role assignment is done or not

```
az cosmosdb sql role assignment exists --account-name $accountName --resource-group
$resourceGroupName --role-assignment-id $role_assignment_id
```

List specific role details

```
az cosmosdb sql role assignment show --account-name $accountName --resource-group
$resourceGroupName --role-assignment-id $role_assignment_id
```

Use this Service principal which has RBAC permissions

- We need to write some dot net code

```
TokenCredential servicePrincipal = new ClientSecretCredential("6bb2f9af-a0af-4c32-a5ec-
5f7011d37551",
"7aae85f5-b903-4220-a8c6-678dc4b73a3f",
"b0.8Q~AL_3pqoPLAqWMxR7MW28GQLWYaP6cVoaEZ");

this.cosmosClient = new CosmosClient(EndpointUri, servicePrincipal);
```