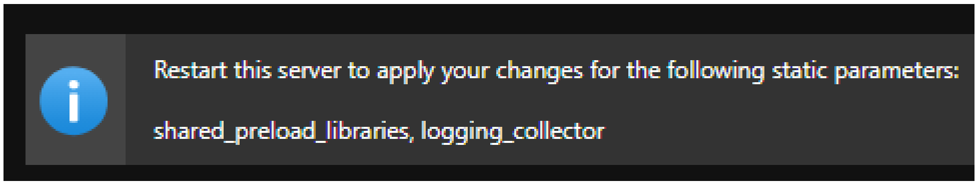
ARM Template - Restart PostgreSQL server after changing a static parameter

Last updated by | Daniel Valero | Jan 21, 2021 at 12:45 PM PST

Introduction

As part of the deployment of a server, it can be necessary to change a static parameter. This can be accomplished with a task of type "Microsoft.DBforPostgreSQL/servers/configurations"

However, for the change to take effect it is necessary to restart the server, otherwise a message like the one in the image below will be displayed in the Portal:



Restart this server to apply your changes for the following static parameters:

shared_preload_libraries, logging_collector

This can create problems on more complex deployments, when after changing a parameter the server must be restarted to expose new parameters, or the deployment will fail when it tries to change not yet exposed parameters. This is the case of changing shared_preload_libraries to enable PgAudit as it is necessary to:

1. add PgAudit to shared_preload_libraries
2. restart the server
3. change PgAudit parameters such as pgaudit.log, pgaudit.log_relation or pgaudit.role

Solution overview

During the deployment, the server can be restarted using a resource of type ***Microsoft.Resources/deploymentScripts*** that executes an Azure CLI command "*az postgres server restart*". This task must depend on the resources that set static parameters

Microsoft.Resources/deploymentScripts requires a Managed Identity to restart the server with the following characteristics:

- The Managed Identity must be assigned to a custom role with at least the permission **Microsoft.DBForPostgreSQL/servers/restart/action**. This is the only permission required to follow the principle of least privilege.
- The role can be created at subscription level, resource group level or resource level.
- The Managed Identity can already exist or can be created as part of the deployment. It can be created in the same resource group that the database server or in another resource group. A Managed Identity can be created for each server, or one can be used per resource group or one for all deployments.
- The Managed Identity and role can already exist or can be created as part of the deployment. To follow the principle of least privilege, the role should exist at resource (database server) level, which means it must be created as part of the deployment.

Solution details

To create the custom role, you can use a resource of type **Microsoft.Authorization/roleDefinitions**

To assign an already existing Managed Identity, you can use a resource of type **Microsoft.DBforPostgreSQL/servers/providers/roleAssignments** that depends on the **Microsoft.Authorization/roleDefinitions** resource

Both the **Microsoft.Authorization/roleDefinitions** and **Microsoft.Authorization/roleDefinitions** resources must be child of the **Microsoft.DBforPostgreSQL/servers** resource and must depend on it.

Below an example, on how to create the server, create the custom role on it and assign an already existing Managed Identity

```
{   "type": "Microsoft.DBforPostgreSQL/servers",
    "apiVersion": "2017-12-01",
    "name": "[parameters('servers_name')]",
    "location": "[parameters('servers_location')]",
    "sku": {
        "name": "GP_Gen5_2",
        "family": "Gen5",
        "capacity": 2
    },
    "properties": {
        "storageProfile": {
            "storageMB": 30720,
            "backupRetentionDays": 35,
            "geoRedundantBackup": "Disabled",
            "storageAutogrow": "Enabled"
        },
        "version": "[parameters('postgresVersion')]",
        "sslEnforcement": "Enabled",
        "minimalTlsVersion": "TLSEnforcementDisabled",
        "infrastructureEncryption": "Disabled",
        "publicNetworkAccess": "Enabled",
        "administratorLogin": "[parameters('postgresAdminLoginName')]",
        "administratorLoginPassword": "[parameters('postgresAdminPassword')]"
    },
                "resources" : [
                        {       "type": "Microsoft.Authorization/roleDefinitions",
                                "apiVersion": "2018-07-01",
                                "name": "[variables('roleDefName')]",
                                "properties": {
                                        "roleName": "[parameters('roleName')]",
                                        "description": "[parameters('roleDescription')]",
                                        "type": "customRole",
                                        "isCustom": true,
                                        "permissions": [
                                                {
                                                "actions": "[parameters('actions')]",
                                                "notActions": "[parameters('notActions')]"
                                                }
                                        ],
                                        "assignableScopes": [
                                          "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters
                                        ]
                                },
                                "dependsOn": [
                                        "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters('
                                ]
                        },
                        {       "type": "Microsoft.DBforPostgreSQL/servers/providers/roleAssignments",
                                "apiVersion": "2020-04-01-preview",
                                "name": "[variables('roleAssignmentName')]",
                                "properties": {
                                        "roleDefinitionId": "[resourceId('Microsoft.Authorization/role
                                        "principalId": "[reference(concat(subscription().id, '/resourc
                                },
                                "dependsOn": [
                                        "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters('
                                        "[resourceId('Microsoft.Authorization/roleDefinitions', variab
                                ]
                        }
                ]
        },
```

In the example below, you can see how to restart the server after changing the *shared_preload_libraries* and *logging_collector* static parameters. The **Microsoft.Resources/deploymentScripts** resource depends on both the two resources **Microsoft.DBforPostgreSQL/servers/configurations**

```
            {       "type": "Microsoft.Resources/deploymentScripts",
                    "apiVersion": "2020-10-01",
                    "name": "RestartAzurePostgreSQLServer",
                    "location": "[resourceGroup().location]",
                    "dependsOn": [
                            "[resourceId('Microsoft.DBforPostgreSQL/servers/configurations', parameters('s
                            "[resourceId('Microsoft.DBforPostgreSQL/servers/configurations', parameters('s
                    ],
                    "identity": {
                            "type": "userAssigned",
                            "userAssignedIdentities": {
                              "[concat(subscription().id, '/resourceGroups/' ,parameters('rg_for_managed_i
                            }
                    },
                    "kind": "AzureCLI",
                    "properties": {
                            "forceUpdateTag": "[parameters('utcValue')]",
                            "AzCliVersion": "2.15.0",
                            "timeout": "PT30M",
                            "arguments": "[concat(parameters('servers_name'), ' ', resourceGroup().name)]"
                            "scriptContent": "az postgres server restart --name $1 --resource-group $2",
                            "cleanupPreference": "Always",
                            "retentionInterval": "PT1H"
                    }
            },
```

Finally, if necessary (as in the case of PgAudit) you change new exposed parameters in the same deployment using **Microsoft.DBforPostgreSQL/servers/configurations** resources that depend on the **Microsoft.Resources/deploymentScripts** resource. An example below on how to change *pgaudit.log* after the server is restarted.

```
        {   "type": "Microsoft.DBforPostgreSQL/servers/configurations",
             "apiVersion": "2017-12-01",
             "name": "[concat(parameters('servers_name'), '/pgaudit.log')]",
             "dependsOn": [
                 "[resourceId('Microsoft.Resources/deploymentScripts', 'RestartAzurePostgreSQLServer')]"
             ],
             "properties": {
                 "value": "role,ddl,read",
                 "source": "user-override"
             }
         },
```

Complete ARM template example

The ARM template below:

1. creates a server
2. creates a custom role on it with **Microsoft.DBForPostgreSQL/servers/restart/action** permission
3. assign an already existing Managed Instance
4. change two static parameters (shared_preload_libraries and logging_collector)
5. change one dynamic parameter (log_statement)
6. restart the server

7. modify parameters exposed after the server is restarted (pgaudit.log and pgaudit.role)

NOTE: The name and resource group of the already existing Managed instance is defined in parameters *name_managed_identity* and *rg_for_managed_identity* respectively. If you will try the template, make sure you change parameters *servers_mame*, *postgresAdminPassword* and any other parameter you need to adjust.

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "servers_name": {
            "defaultValue": "xxxxxxxxxx",
            "type": "String"
        },
        "servers_location": {
            "defaultValue": "eastus2",
            "type": "String"
        },
        "postgresAdminLoginName": {
            "defaultValue": "psql_admin",
            "type": "String"
        },
        "postgresAdminPassword": {
            "defaultValue": "xxxxxxx",
            "type": "String"
        },
        "postgresDefaultExtension": {
            "defaultValue": "timescaledb,pgaudit",
            "type": "String"
        },
        "postgresVersion": {
            "defaultValue": "11",
            "type": "String"
        },
        "utcValue": {
                "type": "string",
                "defaultValue": "[utcNow()]"
        },
        "rg_for_managed_identity": {
                "type": "string",
                "defaultValue": "test1"
        },
        "name_managed_identity": {
                "type": "string",
                "defaultValue": "armdeployuser"
        },
        "actions": {
          "type": "array",
          "defaultValue": [
                "Microsoft.DBForPostgreSQL/servers/restart/action"
          ],
          "metadata": {
                "description": "Array of actions for the roleDefinition"
          }
        },
        "notActions": {
          "type": "array",
          "defaultValue": [],
          "metadata": {
                "description": "Array of notActions for the roleDefinition"
          }
        },
        "roleName": {
          "type": "string",
          "defaultValue": "Custom Role - PostgreSQL Server restart",
          "metadata": {
                "description": "PostgreSQL Server restart only"
          }
        },
        "roleDescription": {
          "type": "string",
          "defaultValue": "Server Level Deployment of a Role Definition",
          "metadata": {
                "description": "Detailed description of the role definition"
          }
        }
  },
  "variables": {
        "roleDefName": "[guid(subscription().id, string(parameters('actions')), string(parameters('notActions'
        "roleAssignmentName": "[concat(parameters('servers_name'), '/Microsoft.Authorization/', guid(subscript
    },
    "resources": [
        {   "type": "Microsoft.DBforPostgreSQL/servers",
            "apiVersion": "2017-12-01",
            "name": "[parameters('servers_name')]",
            "location": "[parameters('servers_location')]",
            "sku": {
                "name": "GP_Gen5_2",
                "family": "Gen5",
                "capacity": 2
            },
            "properties": {
                "storageProfile": {
                    "storageMB": 30720,
                    "backupRetentionDays": 35,
                    "geoRedundantBackup": "Disabled",
                    "storageAutogrow": "Enabled"
                },
                "version": "[parameters('postgresVersion')]",
                "sslEnforcement": "Enabled",
                "minimalTlsVersion": "TLSEnforcementDisabled",
                "infrastructureEncryption": "Disabled",
                "publicNetworkAccess": "Enabled",
                "administratorLogin": "[parameters('postgresAdminLoginName')]",
                "administratorLoginPassword": "[parameters('postgresAdminPassword')]"
            },
                        "resources" : [
                                {       "type": "Microsoft.Authorization/roleDefinitions",
                                        "apiVersion": "2018-07-01",
                                        "name": "[variables('roleDefName')]",
                                        "properties": {
                                                "roleName": "[parameters('roleName')]",
                                                "description": "[parameters('roleDescription')]",
                                                "type": "customRole",
                                                "isCustom": true,
                                                "permissions": [
                                                        {
                                                        "actions": "[parameters('actions')]",
                                                        "notActions": "[parameters('notActions')]"
                                                        }
                                                ],
                                                "assignableScopes": [
                                                  "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters
                                                ]
                                        },
                                        "dependsOn": [
                                                "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters('
                                        ]
                                },
                                {       "type": "Microsoft.DBforPostgreSQL/servers/providers/roleAssignments",
                                        "apiVersion": "2020-04-01-preview",
                                        "name": "[variables('roleAssignmentName')]",
                                        "properties": {
                                                "roleDefinitionId": "[resourceId('Microsoft.Authorization/role
                                                "principalId": "[reference(concat(subscription().id, '/resourc
                                        },
                                        "dependsOn": [
                                                "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters('
                                                "[resourceId('Microsoft.Authorization/roleDefinitions', variab
                                        ]
                                }
                        ]
        },
                {   "type": "Microsoft.DBforPostgreSQL/servers/configurations",
            "apiVersion": "2017-12-01",
            "name": "[concat(parameters('servers_name'), '/shared_preload_libraries')]",
            "dependsOn": [
                "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters('servers_name'))]"
            ],
            "properties": {
                "value": "[parameters('postgresDefaultExtension')]",
                "source": "user-override"
            }
        },
                {       "type": "Microsoft.DBforPostgreSQL/servers/configurations",
            "apiVersion": "2017-12-01",
            "name": "[concat(parameters('servers_name'), '/logging_collector')]",
            "dependsOn": [
                "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters('servers_name'))]"
            ],
            "properties": {
                "value": "OFF",
                "source": "user-override"
            }
        },
                {   "type": "Microsoft.DBforPostgreSQL/servers/configurations",
            "apiVersion": "2017-12-01",
            "name": "[concat(parameters('servers_name'), '/log_statement')]",
            "dependsOn": [
                "[resourceId('Microsoft.DBforPostgreSQL/servers', parameters('servers_name'))]"
            ],
            "properties": {
                "value": "DDL",
                "source": "user-override"
            }
        },
        {       "type": "Microsoft.Resources/deploymentScripts",
                "apiVersion": "2020-10-01",
                "name": "RestartAzurePostgreSQLServer",
                "location": "[resourceGroup().location]",
                "dependsOn": [
                        "[resourceId('Microsoft.DBforPostgreSQL/servers/configurations', parameters('servers_n
                        "[resourceId('Microsoft.DBforPostgreSQL/servers/configurations', parameters('servers_n
                ],
                "identity": {
                        "type": "userAssigned",
                        "userAssignedIdentities": {
                          "[concat(subscription().id, '/resourceGroups/' ,parameters('rg_for_managed_identity'
                        }
                },
                "kind": "AzureCLI",
                "properties": {
                        "forceUpdateTag": "[parameters('utcValue')]",
                        "AzCliVersion": "2.15.0",
                        "timeout": "PT30M",
                        "arguments": "[concat(parameters('servers_name'), ' ', resourceGroup().name)]",
                        "scriptContent": "az postgres server restart --name $1 --resource-group $2",
                        "cleanupPreference": "Always",
                        "retentionInterval": "PT1H"
                }
        },
        {   "type": "Microsoft.DBforPostgreSQL/servers/configurations",
            "apiVersion": "2017-12-01",
            "name": "[concat(parameters('servers_name'), '/pgaudit.log')]",
            "dependsOn": [
                "[resourceId('Microsoft.Resources/deploymentScripts', 'RestartAzurePostgreSQLServer')]"
            ],
            "properties": {
                "value": "role,ddl,read",
                "source": "user-override"
            }
        },
        {   "type": "Microsoft.DBforPostgreSQL/servers/configurations",
            "apiVersion": "2017-12-01",
            "name": "[concat(parameters('servers_name'), '/pgaudit.role')]",
            "dependsOn": [
                "[resourceId('Microsoft.Resources/deploymentScripts', 'RestartAzurePostgreSQLServer')]"
            ],
            "properties": {
                "value": "auditor",
                "source": "user-override"
            }
        }
    ]
}
```

How good have you found this content?

🙂 🙁