DeployIfNotExists

Last updated by | Hamza Aqel | Aug 17, 2022 at 3:09 AM PDT

Contents

DeployIfNotExists

Azure Policy:

Azure Policy helps to enforce organizational standards and to assess compliance at-scale. Through its compliance dashboard, it provides an aggregated view to evaluate the overall state of the environment, with the ability to drill down to the per-resource, per-policy granularity.

Common use cases for Azure Policy include implementing governance for resource consistency, regulatory compliance, security, cost, and management. Policy definitions for these common use cases are already available in your Azure environment as built-ins to help you get started.

One common question is how to make sure that Azure Database for PostgreSQL Flexible Server is compliance through bulk remediation for existing resources and automatic remediation for new resources, use the below Policy sample where it can check a certain server parameter (e.g. log_duration on this case) as DeployIfNotExists:

```
{

  "mode": "All",

  "policyRule": {

    "if": {

      "allOf": [

        {

          "field": "type",

          "equals": "Microsoft.DBforPostgreSQL/flexibleServers"

        }

      ]

    },

    "then": {

      "effect": "[parameters('effect')]",

      "details": {
```

```json
      "type": "Microsoft.DBforPostgreSQL/flexibleServers/configurations",

      "name": "log_duration",

      "existenceCondition": {

        "field": "Microsoft.DBforPostgreSQL/flexibleServers/configurations/value",

        "equals": "ON"

      },

      "roleDefinitionIds": [

        "/providers/microsoft.authorization/roleDefinitions/b24988ac-6180-42a0-ab88-20f7382dd24c"

      ],

      "deployment": {

        "properties": {

          "mode": "incremental",

          "template": {

            "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

            "contentVersion": "1.0.0.0",

            "parameters": {

              "serverName": {

                "type": "string"

              }

            },

            "variables": {},

            "resources": [

              {

                "name": "[concat(parameters('serverName'), '/log_duration')]",

                "type": "Microsoft.DBforPostgreSQL/flexibleServers/configurations",

                "apiVersion": "2022-01-20-preview",

                "properties": {

                  "value": "ON",

                  "source": "user-override"

                }
```
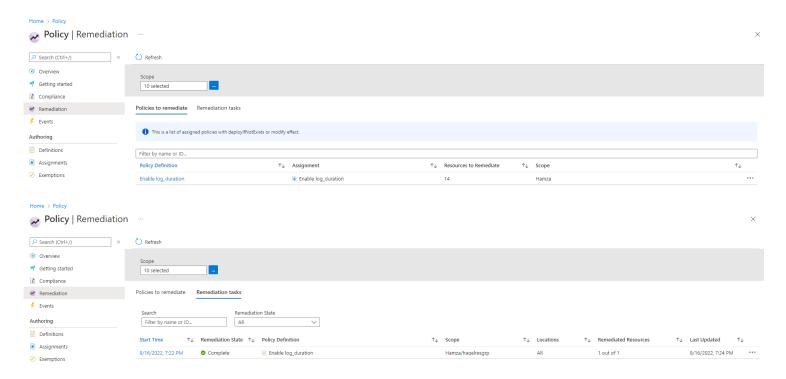
```
                }

              ]

            },

            "parameters": {

              "serverName": {

                "value": "[field('name')]"

              }

            }

          }

        }

      }

    }

  },

  "parameters": {

    "effect": {

      "type": "String",

      "metadata": {

        "displayName": "Effect",

        "description": "Enable or disable the execution of the policy"

      },

      "allowedValues": [

        "DeployIfNotExists",

        "Disabled"

      ],

      "defaultValue": "DeployIfNotExists"

    }

  }

}
```

Notes:

- Based on the above policy all the new resources will have this parameter set to ON by default.

- For existing resources, you need to check and run the Remediation in Azure Policy as below:

For Further use cases for Azure policy, you can share your ideas with

the EEE team / haaqel@micrsoft.com