Create Custom Policies
Policy Sets and Assignments

PREPARED BY :
FAIZAN

In this example you will use three custom policies and a custom policy set definition. The custom policies will be named **Enforce-RG-Tags**, **Enforce-Resource-Tags** and **Deny-NIC-NSG**. You will then create a custom policy set definition (Initiative) named Enforce-Mandatory-Tags that will include the **Enforce-RG-Tags** and **Enforce-Resource-Tags** custom policies.

Steps:-

Here will update the built-in configuration by following these steps:

- Create the custom policy definition file for Enforce-RG-Tags
- Create the custom policy definition file for Enforce-Resource-Tags
- Create the custom policy definition file for Deny-NIC-NSG
- Create the custom policy set definition file for Enforce-Mandatory-Tags
- Make the custom policy definitions available for use in Azure by extending the built-in archetype for es_root
- Create the policy assignment files for Enforce-RG-Tags, Enforce-Resource-Tags, Deny-NIC-NSG and Enforce-Mandatory-Tags
- Assign the custom policy set definition for Enforce-Mandatory-Tags at the es_root Management Group by extending the built-in archetype for es_root
- Assign the custom policy definition for Deny-NIC-NSG at the Landing Zones Management Group by extending the built-in archetype for es_landing_zones

Create Custom Policy Definition

In the policy_definitions subdirectory, create a policy_definition_es_policy_enforce_rg_tags.json file. This file will contain the policy definition for Enforce-RG-Tags

lib/policy_definitions/policy_definition_es_enforce_rg_tags.json

```
{
    "name": "Enforce-RG-Tags",
    "type": "Microsoft.Authorization/policyDefinitions",
    "apiVersion": "2021-06-01",
    "scope": null,
      "properties": {
        "displayName": "Resource groups must have mandatory tagging applied",
        "policyType": "Custom",
        "mode": "All",
        "description": "Enforce mandatory 'Owner' and 'Department' tags on Resource Groups",
        "metadata": {
          "version": "1.0.0",
          "category": "Tags"
        },
        "policyRule": {
        "if": {
            "allOf":[
                {
                "field": "type",
                "equals": "Microsoft.Resources/subscriptions/resourceGroups"
                },
                {
                "anyof": [
                {
                    "field": "[concat('tags[', parameters('Owner'), ']')]",
                    "exists": "false"
                },
                {
                    "field": "[concat('tags[', parameters('Department'), ']')]",
                    "exists": "false"
                }
                ]
            }
```

```
        },
        "then": {
            "effect": "deny"
        }
        },
        "parameters": {
            "Owner": {
                "type": "String",
                "metadata": {
                    "displayName": "Owner",
                    "description": "Specifies the Owner of the Resource Group'"
                }
            },
            "Department": {
                "type": "String",
                "metadata": {
                    "displayName": "Department",
                    "description": "Specifies the Department that the Resource Group belongs to"
                }
            }
        }
    }
}
```

Now create a **policy_definition_es_policy_enforce_resource_tags.**json file. This file will contain the policy definition for Enforce-Resource-Tags.

lib/policy_definitions/policy_definition_es_enforce_resource_tags.json

```
{
"name": "Enforce-Resource-Tags",
"type": "Microsoft.Authorization/policyDefinitions",
"apiVersion": "2021-06-01",
"scope": null,
    "properties": {
    "displayName": "Resources must have mandatory tagging applied",
    "policyType": "Custom",
    "mode": "Indexed",
    "description": "Enforce mandatory 'Owner' and 'Department' tags on Resources",
    "metadata": {
        "version": "1.0.0",
        "category": "Tags"
    },
    "policyRule": {
        "if": {
            "anyof": [
                {
                    "field": "[concat('tags[', parameters('Owner'), ']')]",
                    "exists": "false"
                },
                {
                    "field": "[concat('tags[', parameters('Department'), ']')]",
                    "exists": "false"
                }
                ]
        },
```

```
        "then": {
            "effect": "deny"
        }
    },
        "parameters": {
            "Owner": {
                "type": "String",
                "metadata": {
                    "displayName": "Owner",
                    "description": "Specifies the Owner of the resource"
                }
            },
            "Department": {
                "type": "String",
                "metadata": {
                    "displayName": "Department",
                    "description": "Specifies the Department that the resource belongs to"
                }
            }
        }
    }
}
```

Next create a **policy_definition_es_policy_deny_nsg_nic.json file**. This file will contain the policy definition for our last custom policy - **Deny-NSG-NIC**.

lib/policy_definitions/policy_definition_es_deny_nic_nsg.json

```
{
    "type": "Microsoft.Authorization/policyDefinitions",
    "name": "Deny-NIC-NSG",
    "properties": {
        "displayName": "Prevent Network Security Groups from being applied to Network Interface Cards",
        "description": "This policy will prevent NSGs from being applied to network interface cards.",
        "policyType": "Custom",
        "mode": "All",
        "metadata": {
            "version": "1.0.0",
            "category": "Network"
        },
        "parameters": {
            "effect": {
                "type": "String",
                "defaultValue": "deny",
                "allowedValues": [
                    "audit",
                    "deny",
                    "disabled"
                ],
                "metadata": {
                    "displayName": "Effect",
                    "description": "Enable or disable the execution of the policy"
                }
            }
        },
```

```
        "policyRule": {
            "if": {
                "allOf": [
                {
                        "field": "type",
                        "equals": "Microsoft.Network/networkInterfaces"
                    },
                    {
                        "field": "Microsoft.Network/networkInterfaces/networkSecurityGroup.id",
                        "like": "*"
                    }
                ]
            },
            "then": {
                "effect": "[parameters('effect')]"
            }
        }
    }
}
```

Create Custom Policy Set Definition

In your /lib directory create a policy_set_definitions subdirectory.

In the policy_set_definitions subdirectory, create a policy_set_definition_enforce_mandatory_tags.json file. This file will contain the Policy Set Definition for Enforce-Mandatory-Tags. The policy set will contain the Enforce-RG-Tags and Enforce-Resource-Tags custom policies that you previously created.

lib/policy_set_definitions/policy_set_definition_enforce_mandatory_tagging.json

```
{
    "name": "Enforce-Mandatory-Tags",
    "type": "Microsoft.Authorization/policySetDefinitions",
    "apiVersion": "2021-06-01",
    "scope": null,
    "properties": {
      "policyType": "Custom",
      "displayName": "Ensure mandatory tagging is applied to both Resources and Resource Groups",
      "description": "Contains the core tagging policies applicable to the org",
      "metadata": {
        "version": "1.0.0",
        "category": "General"
      },
      "parameters": {
        "EnforceRGTags-Owner": {
            "type": "String",
            "metadata": {
              "displayName": "Owner",
              "description": "Specifies the Owner of the Resource Group"
            }
        },
```

```
        "EnforceRGTags-Department": {
            "type": "String",
            "metadata": {
              "displayName": "Department",
              "description": "Specifies the Department that the Resource Group belongs to"
            }
        },
        "EnforceResourceTags-Owner": {
            "type": "String",
            "metadata": {
              "displayName": "Owner",
              "description": "Specifies the Owner of the resource"
            }
        },
        "EnforceResourceTags-Department": {
            "type": "String",
            "metadata": {
              "displayName": "Department",
              "description": "Specifies the Department that the resource belongs to"
            }
        }
      },
```

```
      "policyDefinitions": [
        {
          "policyDefinitionReferenceId": "Resource groups must have mandatory tagging applied",
          "policyDefinitionId": "${root_scope_resource_id}/providers/Microsoft.Authorization/policyDefinitions/Enforce
          "parameters": {
            "Owner": {
              "value": "[parameters('EnforceRGTags-Owner')]"
            },
            "Department": {
              "value": "[parameters('EnforceRGTags-Department')]"
            }
          },
          "groupNames": []
        },
        {
          "policyDefinitionReferenceId": "Resources must have mandatory tagging applied",
          "policyDefinitionId": "${root_scope_resource_id}/providers/Microsoft.Authorization/policyDefinitions/Enforce
          "parameters": {
            "Owner": {
              "value": "[parameters('EnforceResourceTags-Owner')]"
            },
            "Department": {
              "value": "[parameters('EnforceResourceTags-Department')]"
            }
          },
```

```
          "groupNames": []
        }
      ],
      "policyDefinitionGroups": null
    }
  }
```

Create Custom Policy Assignment Files

In order to assign your custom policies or policy sets, you need to create policy assignment files. The first step is to create a policy_assignments subdirectory within /lib.

Here will then need to create a file named policy_assignment_es_enforce_rg_tags.json within the policy_assignments directory.

lib/policy_assignments/policy_assignment_es_enforce_rg_tags.json

```
{
    "name": "Enforce-RG-Tags",
    "type": "Microsoft.Authorization/policyAssignments",
    "apiVersion": "2019-09-01",
    "properties": {
        "description": "Enforce Mandatory Tags on Resource Groups",
        "displayName": "Resource groups must have mandatory tagging applied",
        "notScopes": [],
        "parameters": {
        } ,
        "policyDefinitionId": "${root_scope_resource_id}/providers/Microsoft.Authorization/policyDefinitions/Enforce-R
        "nonComplianceMessages": [
          {
            "message": "Mandatory tags {enforcementMode} be applied to Resource Groups."
          }
        ],
        "scope": "${current_scope_resource_id}",
        "enforcementMode": null,
        "nonComplianceMessages": [
        {
            "message": "Mandatory tags must be provided."
        }
        ]
    },
```

```
        "scope": "${current_scope_resource_id}",
        "enforcementMode": null,
        "nonComplianceMessages": [
        {
            "message": "Mandatory tags must be provided."
        }
        ]
    },
    "location": "${default_location}",
    "identity": {
        "type": "SystemAssigned"
    }
}
```

Now create a file named policy_assignment_es_enforce_resource_tags.json within the policy_assignments directory

lib/policy_assignments/policy_assignment_es_enforce_resource_tags.json

```
{
    "name": "Enforce-Resource-Tags",
    "type": "Microsoft.Authorization/policyAssignments",
    "apiVersion": "2019-09-01",
    "properties": {
        "description": "Enforce Mandatory Tags on Resources",
        "displayName": "Resources must have mandatory tagging applied",
        "notScopes": [],
        "parameters": {
        } ,
        "policyDefinitionId": "${root_scope_resource_id}/providers/Microsoft.Authorization/policyDefinitions/Enforce-
        "nonComplianceMessages": [
          {
            "message": "Mandatory tags {enforcementMode} be applied to resources."
          }
        ],
        "scope": "${current_scope_resource_id}",
        "enforcementMode": null,
        "nonComplianceMessages": [
        {
            "message": "Mandatory tags must be provided."
        }
        ]
    },
    "location": "${default_location}",
    "identity": {
        "type": "SystemAssigned"
    }
}
```

Next create a file named policy_assignment_es_deny_nic_nsg.json within the policy_assignments directory.

lib/policy_assignments/policy_assignment_es_deny_nic_nsg.json

```
{
    "name": "Deny-NIC-NSG",
    "type": "Microsoft.Authorization/policyAssignments",
    "apiVersion": "2019-09-01",
    "properties": {
      "description": "This policy will prevent NSGs from being applied to network interface cards.",
      "displayName": "Prevent Network Security Groups from being applied to Network Interface Cards",
      "notScopes": [],
      "parameters": {} ,
      "policyDefinitionId": "${root_scope_resource_id}/providers/Microsoft.Authorization/policyDefinitions/Deny-NIC-NS
      "nonComplianceMessages": [
          {
            "message": "NSGs {enforcementMode} not be applied to network interface cards."
          }
        ],
      "scope": "${current_scope_resource_id}",
      "enforcementMode": null,
      "nonComplianceMessages": [
        {
            "message": "NSGs must not be applied to Network Interface cards."
        }
    ]

    },
    "location": "${default_location}",
    "identity": {
      "type": "SystemAssigned"
    }
  }
```

Finally, create a file named policy_assignment_es_enforce_mandatory_tagging.json.

lib/policy_assignments/policy_assignment_es_enforce_mandatory_tagging.json

```
{
  "name": "Enforce-Mandatory-Tags",
  "type": "Microsoft.Authorization/policyAssignments",
  "apiVersion": "2019-09-01",
  "properties": {
    "description": "Contains the core policies applicable to the org",
    "displayName": "Ensure mandatory tagging is applied to both Resources and Resource Groups",
    "notScopes": [],
    "parameters": {
      "EnforceRGTags-Owner": {
        "Value": "Jane Doe"
      },
      "EnforceRGTags-Department": {
        "Value": "IT"
      },
      "EnforceResourceTags-Owner": {
        "Value": "Jane Doe"
      },
      "EnforceResourceTags-Department": {
        "Value": "IT"
      }
    },
    "policyDefinitionId": "${root_scope_resource_id}/providers/Microsoft.Authorization/policySetDefinitions/Enforce-Ma
    "nonComplianceMessages": [
      {
        "message": "Mandatory tags {enforcementMode} be applied to Resources and Resource Groups."
      }
    ],
    "scope": "${current_scope_resource_id}",
    "enforcementMode": null
  },
  "location": "${default_location}",
  "identity": {
    "type": "SystemAssigned"
  }
}
```

Assign the Enforce-Mandatory-Tags Custom Policy Set at the es_root Management Group.

You now need to assign the Enforce-Mandatory-Tags policy set and in this example, we will assign it at es_root. To do this, update your existing archetype_extension_es_root.tmpl.json

```
{
  "extend_es_root": {
    "policy_assignments": ["Enforce-Mandatory-Tags"],
    "policy_definitions": ["Enforce-RG-Tags", "Enforce-Resource-Tags", "Deny-NIC-NSG"],
    "policy_set_definitions": ["Enforce-Mandatory-Tags"],
    "role_definitions": [],
    "archetype_config": {
      "access_control": {
      }
    }
  }
}
```

You have now successfully created and assigned both a Custom Policy Definition and a Custom Policy Set Definition within your Azure environment. You can re-use the steps in this article for any Custom Policies of your own that you may wish to use.