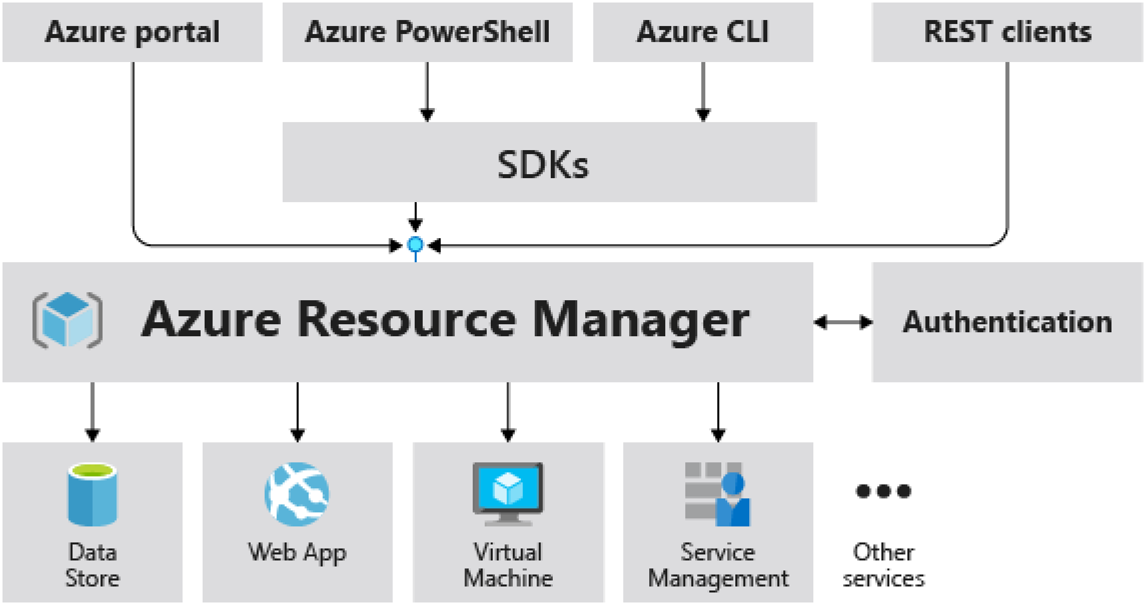
Deploy Management Resources With Custom Settings.
(Hub & Spoke Level 300)

By:
Sarfaraz Momin
(Cloud Engineer)

What is Management Resources?

Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account. You use management features, like access control, locks, and tags, to secure and organize your resources after deployment.



Strategy

we take the base Deploy Management resources configuration and make the following changes:

- Add input variable on the root module for enabling/disabling Management resources
- Add a local variable for configure_management_resources and set custom values for the following:
    - Update the retention period for data stored in the Log Analytics workspace from 30 days to 50 days (controlled through an input variable on the root module)
    - Set a valid email address for Security alerts (controlled through an input variable on the root module)
    - Disable Azure Defender for Azure Kubernetes Service (AKS)
    - Set a different location for Management resources (controlled through an input variable on the root module)
    - Add custom resource tags for Management resources (controlled through an input variable on the root module)
    - Disable deployment of specified monitoring solutions in Azure Monitor (ServiceMap, SQLAssessment, SQLAdvancedThreatProtection, SQLVulnerabilityAssessment)

The module allows for further customization of the Management resources through the advanced setting, however this is out-of-scope for this example.

If you've already deployed the Management resources using default settings, you will be able to see the changes made when moving to this configuration.

If location is not specified, the resources will default to the same location set by default_location input variable.

Root Module:

To make the code easier to maintain when extending your configuration, we recommend splitting the root module into multiple files. For the purpose of this example, we use the following:

- terraform.tf
- variables.tf
- main.tf
- settings.connectivity.tf

▪ terraform.tf

```
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.54.0"
    }
  }
}

provider "azurerm" {
  features {}
}
```

variables.tf

```
# Use variables to customize the deployment

variable "root_id" {
  type    = string
  default = "myorg"
}

variable "root_name" {
  type    = string
  default = "My Organization"
}

variable "deploy_management_resources" {
  type    = bool
  default = true
}

variable "log_retention_in_days" {
  type    = number
  default = 50
}

variable "security_alerts_email_address" {
  type    = string
  default = "mominsarfraz6677@gmail.com" # Replace this value with your own email address.
}

variable "management_resources_location" {
  type    = string
  default = "uksouth"
}

variable "management_resources_tags" {
  type = map(string)
  default = {
    demo_type = "deploy_management_resources_custom"
  }
}
```

- **main.tf**

```
# Get the current client configuration from the AzureRM provider.
# This is used to populate the root_parent_id variable with the
# current Tenant ID used as the ID for the "Tenant Root Group"
# Management Group.

data "azurerm_client_config" "core" {}

# Declare the Azure landing zones Terraform module
# and provide a base configuration.

module "enterprise_scale" {
  source  = "Azure/caf-enterprise-scale/azurerm"
  version = "4.0.1" # change this to your desired
version,https://www.terraform.io/language/expressions/version-constraints

  default_location = "eastus"

  providers = {
    azurerm              = azurerm
    azurerm.connectivity = azurerm
    azurerm.management   = azurerm
  }

  root_parent_id = data.azurerm_client_config.core.tenant_id
  root_id        = var.root_id
  root_name      = var.root_name

  deploy_management_resources    = var.deploy_management_resources
  subscription_id_management     = data.azurerm_client_config.core.subscription_id
  configure_management_resources = local.configure_management_resources
}
```

Please edit `version = "<VERSION>"` **&** `default_location = "YOUR_LOCATION"`

- **settings.connectivity.tf**

```
# Configure the management resources settings.
locals {
  configure_management_resources = {
    settings = {
      log_analytics = {
        enabled = true
        config = {
          retention_in_days                                 = var.log_retention_in_days
          enable_monitoring_for_vm                          = true
          enable_monitoring_for_vmss                        = true
          enable_solution_for_agent_health_assessment       = true
          enable_solution_for_anti_malware                  = true
          enable_solution_for_change_tracking               = true
          enable_solution_for_service_map                   = false
          enable_solution_for_sql_assessment                = false
          enable_solution_for_sql_vulnerability_assessment  = false
          enable_solution_for_sql_advanced_threat_detection = false
          enable_solution_for_updates                       = true
          enable_solution_for_vm_insights                   = true
          enable_sentinel                                   = true
        }
      }
      security_center = {
        enabled = true
        config = {
          email_security_contact             = var.security_alerts_email_address
          enable_defender_for_app_services   = true
          enable_defender_for_arm            = true
          enable_defender_for_containers     = false
          enable_defender_for_dns            = true
          enable_defender_for_key_vault      = true
          enable_defender_for_oss_databases  = true
          enable_defender_for_servers        = true
          enable_defender_for_sql_servers    = true
          enable_defender_for_sql_server_vms = true
          enable_defender_for_storage        = true
        }
      }
    }

    location = var.management_resources_location
    tags     = var.management_resources_tags
    advanced = null
  }
}
```

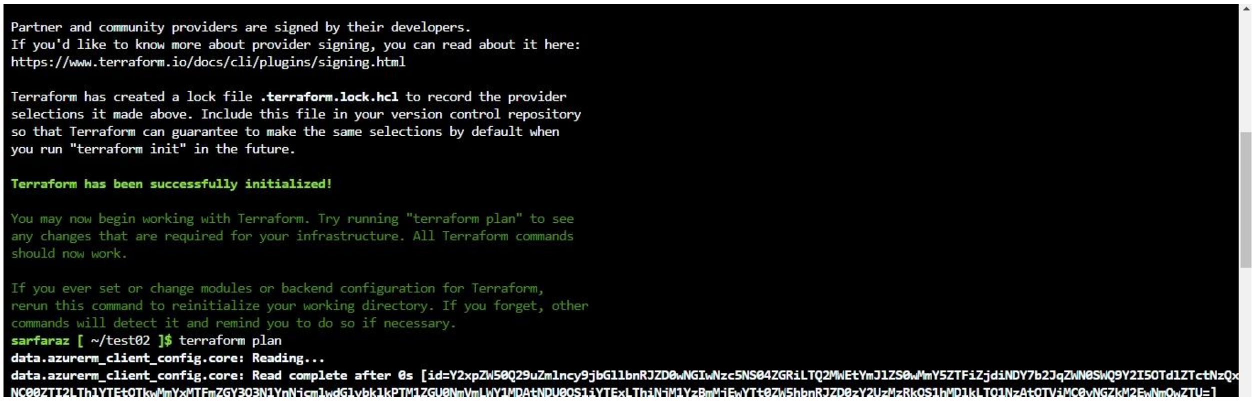
Number of Policies in This Deployment.

Count: 336

Number of Resources in This Deployment.

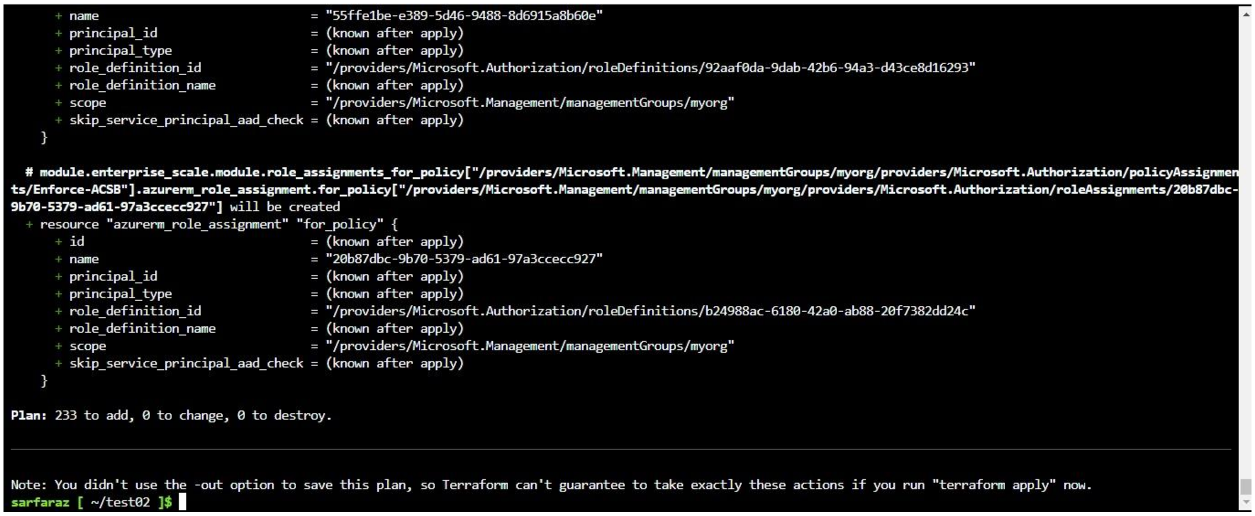
Count: 12

Terraform init:



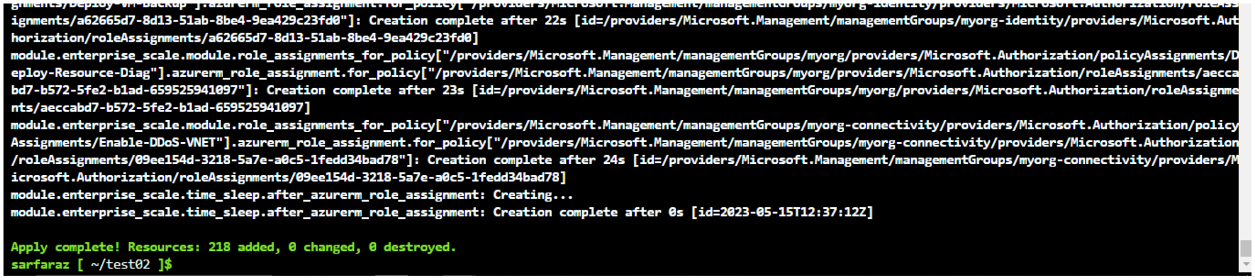
```
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
sarfaraz [ ~/test02 ]$ terraform plan
data.azurerm_client_config.core: Reading...
data.azurerm_client_config.core: Read complete after 0s [id=Y2xpZW50Q29uZmlncy9jbGllbnRJZD0wNGIwNzc5NS04ZGRiLTQ2MWEtYmJlZS0wMmY5ZTFiZjdiNDY7b2JqZWN0SWQ9Y2I5OTdlZTctNzQx
NC00ZTI2LTh1YTEtOTkwMmYxMTFmZGY3O3N1YnNjcmlwdGlvbklkPTM1ZGU0NmVmLWY1MDAtNDU0OS1iYTExLThiNjM1YzBmMjEwYTt0ZW5hbnRJZD0zY2UzMzRkOS1hMDlkLTQ1NzAtOTViMC0yNGZkM2EwNmQwZTU=]
```

Terraform plan:



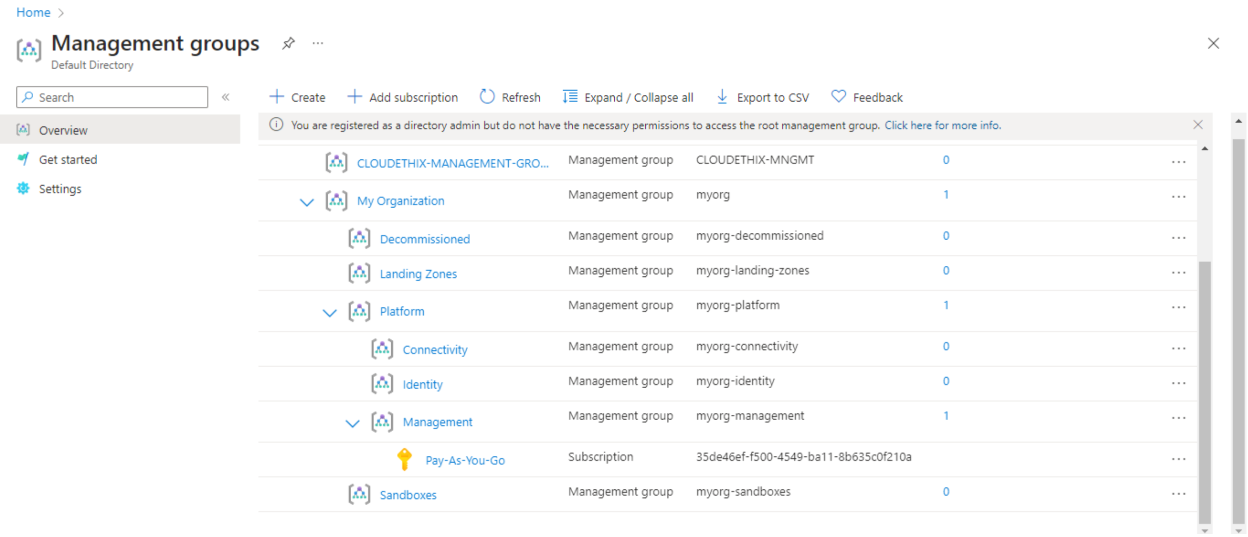
```
      + name                            = "55ffe1be-e389-5d46-9488-8d6915a8b60e"
      + principal_id                    = (known after apply)
      + principal_type                  = (known after apply)
      + role_definition_id              = "/providers/Microsoft.Authorization/roleDefinitions/92aaf0da-9dab-42b6-94a3-d43ce8d16293"
      + role_definition_name            = (known after apply)
      + scope                           = "/providers/Microsoft.Management/managementGroups/myorg"
      + skip_service_principal_aad_check = (known after apply)
    }

  # module.enterprise_scale.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg/providers/Microsoft.Authorization/policyAssignmen
ts/Enforce-ACSB"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg/providers/Microsoft.Authorization/roleAssignments/20b87dbc-
9b70-5379-ad61-97a3ccecc927"] will be created
  + resource "azurerm_role_assignment" "for_policy" {
      + id                              = (known after apply)
      + name                            = "20b87dbc-9b70-5379-ad61-97a3ccecc927"
      + principal_id                    = (known after apply)
      + principal_type                  = (known after apply)
      + role_definition_id              = "/providers/Microsoft.Authorization/roleDefinitions/b24988ac-6180-42a0-ab88-20f7382dd24c"
      + role_definition_name            = (known after apply)
      + scope                           = "/providers/Microsoft.Management/managementGroups/myorg"
      + skip_service_principal_aad_check = (known after apply)
    }

Plan: 233 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
sarfaraz [ ~/test02 ]$
```

Terraform apply:



```
gnments/Deploy-VM-Backup"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-identity/providers/Microsoft.Authorization/roleAss
ignments/a62665d7-8d13-51ab-8be4-9ea429c23fd0"]: Creation complete after 22s [id=/providers/Microsoft.Management/managementGroups/myorg-identity/providers/Microsoft.Aut
horization/roleAssignments/a62665d7-8d13-51ab-8be4-9ea429c23fd0]
module.enterprise_scale.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg/providers/Microsoft.Authorization/policyAssignments/D
eploy-Resource-Diag"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg/providers/Microsoft.Authorization/roleAssignments/aecca
bd7-b572-5fe2-b1ad-659525941097"]: Creation complete after 23s [id=/providers/Microsoft.Management/managementGroups/myorg/providers/Microsoft.Authorization/roleAssignme
nts/aeccabd7-b572-5fe2-b1ad-659525941097]
module.enterprise_scale.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-connectivity/providers/Microsoft.Authorization/policy
Assignments/Enable-DDoS-VNET"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-connectivity/providers/Microsoft.Authorization
/roleAssignments/09ee154d-3218-5a7e-a0c5-1fedd34bad78"]: Creation complete after 24s [id=/providers/Microsoft.Management/managementGroups/myorg-connectivity/providers/M
icrosoft.Authorization/roleAssignments/09ee154d-3218-5a7e-a0c5-1fedd34bad78]
module.enterprise_scale.time_sleep.after_azurerm_role_assignment: Creating...
module.enterprise_scale.time_sleep.after_azurerm_role_assignment: Creation complete after 0s [id=2023-05-15T12:37:12Z]

Apply complete! Resources: 218 added, 0 changed, 0 destroyed.
sarfaraz [ ~/test02 ]$
```

Deployed Management Groups

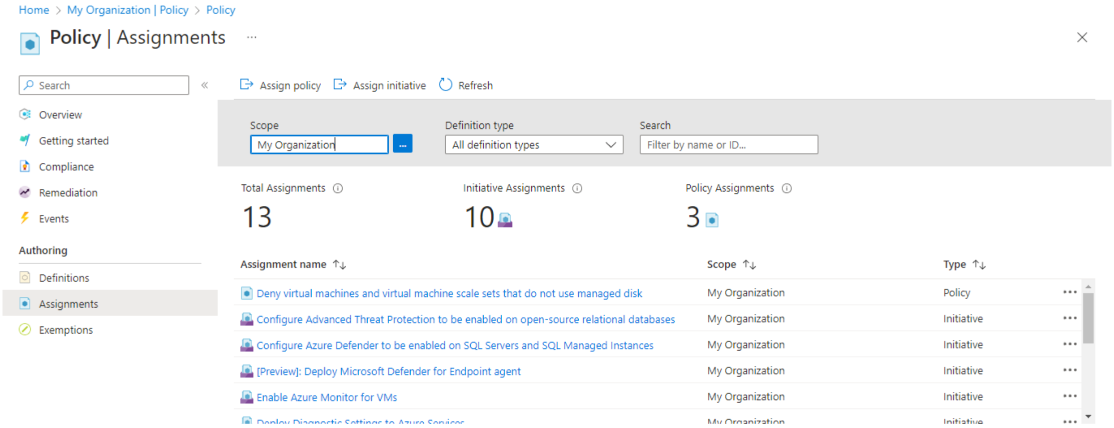


ⓘ You are registered as a directory admin but do not have the necessary permissions to access the root management group. Click here for more info.

CLOUDETHIX-MANAGEMENT-GRO...	Management group	CLOUDETHIX-MNGMT	0	...
My Organization	Management group	myorg	1	...
Decommissioned	Management group	myorg-decommissioned	0	...
Landing Zones	Management group	myorg-landing-zones	0	...
Platform	Management group	myorg-platform	1	...
Connectivity	Management group	myorg-connectivity	0	...
Identity	Management group	myorg-identity	0	...
Management	Management group	myorg-management	1	...
Pay-As-You-Go	Subscription	35de46ef-f500-4549-ba11-8b635c0f210a		...
Sandboxes	Management group	myorg-sandboxes	0	...

Policy Assignment configuration

Policy Assignment parameters



Home > My Organization | Policy > Policy

Policy | Assignments ···

Search «

- Overview
- Getting started
- Compliance
- Remediation
- Events

Authoring

- Definitions
- Assignments
- Exemptions

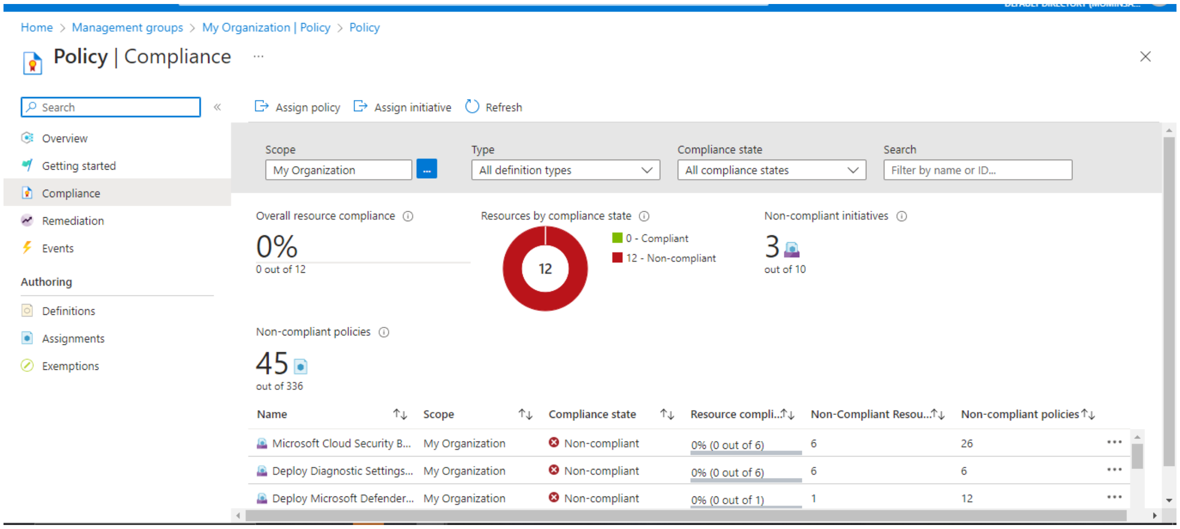
Assign policy · Assign initiative · Refresh

Scope: My Organization · Definition type: All definition types · Search: Filter by name or ID...

Total Assignments: 13 · Initiative Assignments: 10 · Policy Assignments: 3

Assignment name ↑↓	Scope ↑↓	Type ↑↓
Deny virtual machines and virtual machine scale sets that do not use managed disk	My Organization	Policy
Configure Advanced Threat Protection to be enabled on open-source relational databases	My Organization	Initiative
Configure Azure Defender to be enabled on SQL Servers and SQL Managed Instances	My Organization	Initiative
[Preview]: Deploy Microsoft Defender for Endpoint agent	My Organization	Initiative
Enable Azure Monitor for VMs	My Organization	Initiative
Deploy Diagnostic Settings to Azure Services	My Organization	Initiative

Policy Assignment compliance



Home > Management groups > My Organization | Policy > Policy

Policy | Compliance ···

Search «

- Overview
- Getting started
- Compliance
- Remediation
- Events

Authoring

- Definitions
- Assignments
- Exemptions

Assign policy · Assign initiative · Refresh

Scope: My Organization · Type: All definition types · Compliance state: All compliance states · Search: Filter by name or ID...

Overall resource compliance: 0% (0 out of 12)

Resources by compliance state: 12

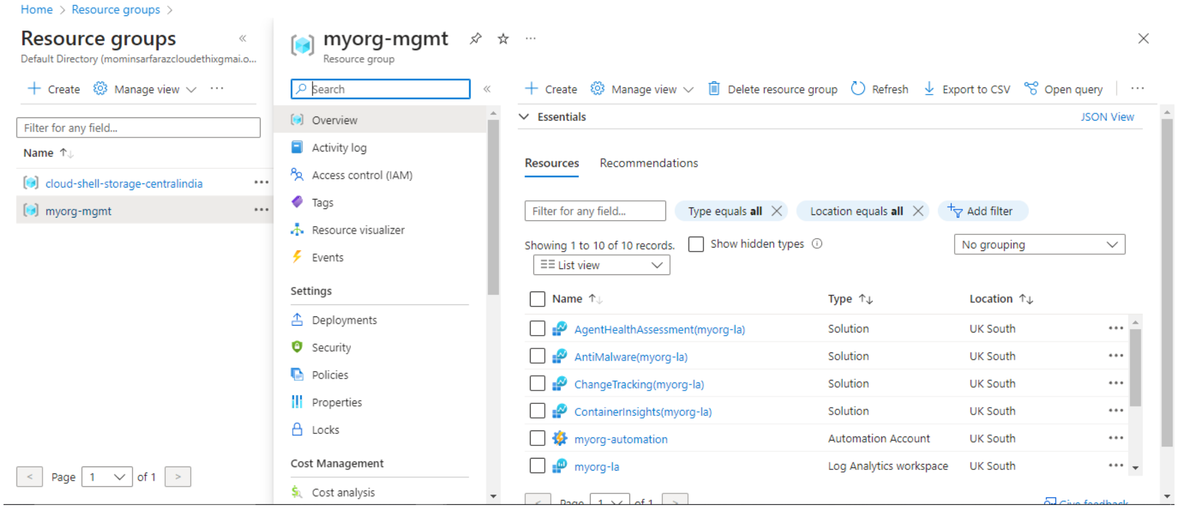
- 🟩 0 - Compliant
- 🟥 12 - Non-compliant

Non-compliant initiatives: 3 out of 10

Non-compliant policies: 45 out of 336

Name ↑↓	Scope ↑↓	Compliance state ↑↓	Resource compli...↑↓	Non-Compliant Resou...↑↓	Non-compliant policies ↑↓
Microsoft Cloud Security B...	My Organization	Non-compliant	0% (0 out of 6)	6	26
Deploy Diagnostic Settings...	My Organization	Non-compliant	0% (0 out of 6)	6	6
Deploy Microsoft Defender...	My Organization	Non-compliant	0% (0 out of 1)	1	12

Deployed Management resources



Home > Resource groups >

Resource groups «

Default Directory (mominsarfarazcloudethixgmai.o...

+ Create · Manage view ···

Filter for any field...

Name ↑↓
cloud-shell-storage-centralindia
myorg-mgmt

< Page 1 of 1 >

myorg-mgmt ···

Resource group

Search «

- Overview
- Activity log
- Access control (IAM)
- Tags
- Resource visualizer
- Events

Settings

- Deployments
- Security
- Policies
- Properties
- Locks

Cost Management

- Cost analysis

+ Create · Manage view · Delete resource group · Refresh · Export to CSV · Open query ···

Essentials · JSON View

Resources · Recommendations

Filter for any field... · Type equals **all** · Location equals **all** · Add filter

Showing 1 to 10 of 10 records. □ Show hidden types · No grouping · List view

□ Name ↑↓	Type ↑↓	Location ↑↓
□ AgentHealthAssessment(myorg-la)	Solution	UK South
□ AntiMalware(myorg-la)	Solution	UK South
□ ChangeTracking(myorg-la)	Solution	UK South
□ ContainerInsights(myorg-la)	Solution	UK South
□ myorg-automation	Automation Account	UK South
□ myorg-la	Log Analytics workspace	UK South

Additional considerations

If you are using Archetype Exclusions or custom Archetypes in your code, make sure to not disable Log Analytics or Security Center policies if you require policy integration using this module. The relationship between the resources deployed and the Policy parameters is dependent on specific Policy Assignments being used.

Next steps

Take particular note of the following changes:

- **The retentionInDays setting is now configured to 50 days on the Log Analytics workspace.**
- **The dataRetention parameter value is also configured to 50 days on the Deploy-Log-Analytics Policy Assignment.**
- **The emailSecurityContact parameter value is set to your own email address on the Deploy-MDFC-Config (Deploy Azure Security Center configuration) Policy Assignment. Once this policy is remediated, you can also view this setting in Azure Security Center.**
- **The pricingTierKubernetesService parameter value is set to Free on the Deploy-MDFC-Config (Deploy Azure Security Center configuration) Policy Assignment. In Security Center, you should be able to see that Azure Defender is set to On for all resource types except Kubernetes 1 which is set to Off.**

Although not Policy Assignment related, also note the following changes:

- **All Resource Groups and Resources created by the module for Management are now located in uksouth.**
- **All Resource Groups and Resources (which support tags) created by the module for Management have the custom tags applied.**

Try updating the configuration settings in the configure_management_resources local variable to see how this changes your configuration. Also try setting your own values in the input variables, and toggling the deploy_management_resources input variable to see which resources are created/destroyed.