Deploy using multiple module declarations with orchestration

By:
Sarfaraz Momin & Kuldip Gund
(Cloud Engineer)

Strategy

This is example of how you could deploy your Azure landing zone using multiple declarations of the module with an orchestration module to manage deployment within a single Terraform workspace.

When segregating a deployment across multiple module instances, it's important to understand how the module works and what inputs are needed. All resources are created based on a data model which uses the configuration inputs to determine certain values.

These values are then shared across the different child modules to determine which resources to create, and how to configure policies. Feature flags such as deploy_connectivity_resources are then used to control whether the module actually creates the resources, or just builds the data model for policy. As such, it's important to keep consistent inputs across each module instance when separating capabilities across different module instances.

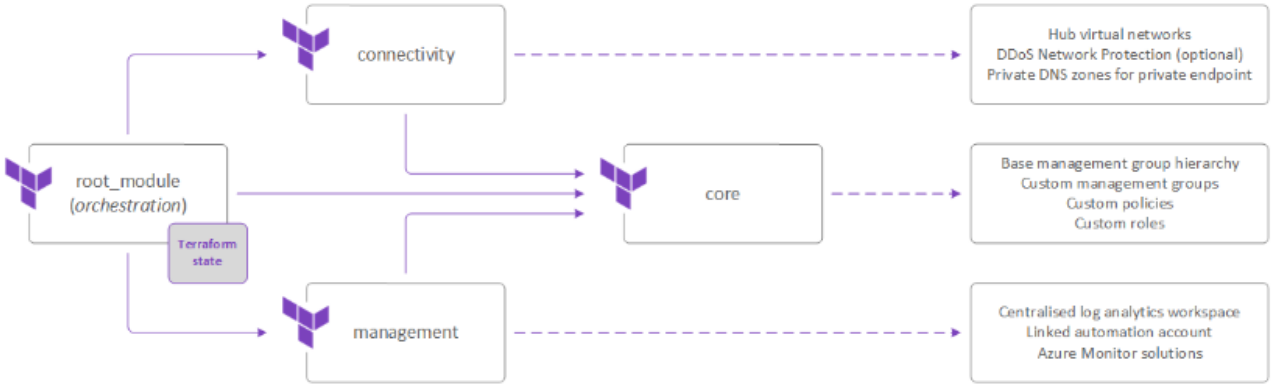
This is demonstrated in this example by the root (orchestration) module which ensure that the core module instance is populated with the same configuration data (by scope) as the management and connectivity modules instances.

This example builds on top of existing examples, including:

- **Deploy Custom Landing Zone Archetypes**
- **Deploy connectivity resources with custom settings**
- **Deploy Management Resources With Custom Settings**

Module composition:

This example is composed of the following modules in a nested structure:



Splitting the code across the following files (grouped by folder for each child module):

root_module/

- **main.tf**
- **variables.tf**
- ❖ **modules/**
  - **connectivity/**
  - **main.tf**
  - **outputs.tf**
  - **settings.connectivity.tf**
  - **variables.tf**
- ❖ **core/**
  - **main.tf**
  - **settings.core.tf**
  - **settings.identity.tf**
  - **variables.tf**
- ❖ **management/**
  - **main.tf**
  - **outputs.tf**
  - **settings.management.tf**
  - **variables.tf**

--CODE--

main.tf

The main.tf file is used as an orchestration module, defining references to multiple instances of the Azure landing zones Terraform module for connectivity, management and core resources. To simplify the example, it also includes code to set the provider configuration, including pinning to a specific version (or range of versions) for the AzureRM Provider. For production use, we recommend pinning to a specific version, and not using ranges.

This example includes logic allowing use of either a single or multiple platform Subscriptions for connectivity and management resources. If an identity Subscription is specified, this will be moved to the identity management group but no resources will be deployed to this Subscription.

```
# Configure Terraform to set the required AzureRM provider
# version and features{} block

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

# Define the provider configuration

provider "azurerm" {
  features {}
}

# Get the current client configuration from the AzureRM provider

data "azurerm_client_config" "current" {}

# Logic to handle 1-3 platform subscriptions as available

locals {
  subscription_id_connectivity = coalesce(var.subscription_id_connectivity,
local.subscription_id_management)
  subscription_id_identity     = coalesce(var.subscription_id_identity,
local.subscription_id_management)
  subscription_id_management   = coalesce(var.subscription_id_management,
data.azurerm_client_config.current.subscription_id)
}
```

```
# The following module declarations act to orchestrate the
# independently defined module instances for core,
# connectivity and management resources

module "connectivity" {
  source = "./modules/connectivity"

  connectivity_resources_tags  = var.connectivity_resources_tags
  enable_ddos_protection       = var.enable_ddos_protection
  primary_location             = var.primary_location
  root_id                      = var.root_id
  secondary_location           = var.secondary_location
  subscription_id_connectivity = local.subscription_id_connectivity
}

module "management" {
  source = "./modules/management"

  email_security_contact     = var.email_security_contact
  log_retention_in_days      = var.log_retention_in_days
  management_resources_tags  = var.management_resources_tags
  primary_location           = var.primary_location
  root_id                    = var.root_id
  subscription_id_management = local.subscription_id_management
}

module "core" {
  source = "./modules/core"

  configure_connectivity_resources = module.connectivity.configuration
  configure_management_resources   = module.management.configuration
  primary_location                 = var.primary_location
  root_id                          = var.root_id
  root_name                        = var.root_name
  secondary_location               = var.secondary_location
  subscription_id_connectivity     = local.subscription_id_connectivity
  subscription_id_identity         = local.subscription_id_identity
  subscription_id_management       = local.subscription_id_management
}
```

variables.tf

The variables.tf file is used to declare a couple of example variables which are used to customize deployment of this root module across all capabilities. Defaults are provided for simplicity, but these should be replaced or over-ridden with values suitable for your environment.

```
# Use variables to customize the deployment

variable "root_id" {
  type        = string
  description = "Sets the value used for generating unique resource naming within the
module."
  default     = "myorg"
}

variable "root_name" {
  type        = string
  description = "Sets the value used for the \"intermediate root\" management group
display name."
  default     = "My Organization"
}

variable "primary_location" {
  type        = string
  description = "Sets the location for \"primary\" resources to be created in."
  default     = "northeurope"
}

variable "secondary_location" {
  type        = string
  description = "Sets the location for \"secondary\" resources to be created in."
  default     = "westeurope"
}

variable "subscription_id_connectivity" {
  type        = string
  description = "Subscription ID to use for \"connectivity\" resources."
  default     = ""
}

variable "subscription_id_identity" {
  type        = string
  description = "Subscription ID to use for \"identity\" resources."
  default     = ""
}
```

```
variable "subscription_id_management" {
  type        = string
  description = "Subscription ID to use for \"management\" resources."
  default     = ""
}

variable "email_security_contact" {
  type        = string
  description = "Set a custom value for the security contact email address."
  default     = "test.user@replace_me"
}

variable "log_retention_in_days" {
  type        = number
  description = "Set a custom value for how many days to store logs in the Log Analytics
workspace."
  default     = 60
}

variable "enable_ddos_protection" {
  type        = bool
  description = "Controls whether to create a DDoS Network Protection plan and link to
hub virtual networks."
  default     = false
}

variable "connectivity_resources_tags" {
  type        = map(string)
  description = "Specify tags to add to \"connectivity\" resources."
  default = {
    deployedBy = "terraform/azure/caf-enterprise-scale/examples/l400-multi"
    demo_type  = "Deploy connectivity resources using multiple module declarations"
  }
}

variable "management_resources_tags" {
  type        = map(string)
  description = "Specify tags to add to \"management\" resources."
  default = {
    deployedBy = "terraform/azure/caf-enterprise-scale/examples/l400-multi"
    demo_type  = "Deploy management resources using multiple module declarations"
  }
}
```

modules/connectivity/main.tf

The modules/connectivity/main.tf file contains a customized module declaration to create two hub networks and DNS resources in your connectivity Subscription.

It also includes the necessary Terraform and provider configuration, and an azurerm_client_config resource which is used to determine the Tenant ID and Subscription ID values for the context being used to create these resources. This is used to ensure the deployment will target your Tenant Root Group by default, and to populate the subscription_id_connectivity input variable.

```
# Configure Terraform to set the required AzureRM provider
# version and features{} block

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

# Define the provider configuration

provider "azurerm" {
  features {}

  subscription_id = var.subscription_id_connectivity
}

# Get the current client configuration from the AzureRM provider

data "azurerm_client_config" "current" {}

# Declare the Azure landing zones Terraform module
# and provide the connectivity configuration

module "alz" {
  source  = "Azure/caf-enterprise-scale/azurerm"
  version = "<version>" # 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
  }

  # Base module configuration settings
  root_parent_id = data.azurerm_client_config.current.tenant_id
  root_id        = var.root_id

  # Disable creation of the core management group hierarchy
  # as this is being created by the core module instance
  deploy_core_landing_zones = false

  # Configuration settings for connectivity resources
  deploy_connectivity_resources    = true
  configure_connectivity_resources = local.configure_connectivity_resources
  subscription_id_connectivity     = var.subscription_id_connectivity

}
```

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

modules/connectivity/outputs.tf

The modules/connectivity/outputs.tf file contains modules outputs used when connecting the module instances together.

The configuration output is an important part of this example, as this is used to ensure the same values used to configure the connectivity resources is shared with the core module instance. This ensures that managed parameters for policies deployed by the core module instance are configured with values correctly reflecting the resources deployed by this module instance.

```
# Output a copy of configure_connectivity_resources for use
# by the core module instance

output "configuration" {
  description = "Configuration settings for the \"connectivity\" resources."
  value       = local.configure_connectivity_resources
}
```

modules/connectivity/settings.connectivity.tf

The modules/connectivity/settings.connectivity.tf file is used to specify the configuration used for creating the required connectivity resources.

This is used as an input to the connectivity module instance, but is also shared with the core module instance to ensure consistent configuration between resources and policies.

```
# Configure custom connectivity resources settings
locals {
  configure_connectivity_resources = {
    settings = {
      # Create two hub networks with hub mesh peering enabled
      # and link to DDoS protection plan if created
      hub_networks = [
        {
          config = {
            address_space                  = ["10.100.0.0/22", ]
            location                       = var.primary_location
            link_to_ddos_protection_plan   = var.enable_ddos_protection
            enable_hub_network_mesh_peering = true
          }
        },
        {
          config = {
            address_space                  = ["10.101.0.0/22", ]
            location                       = var.secondary_location
            link_to_ddos_protection_plan   = var.enable_ddos_protection
            enable_hub_network_mesh_peering = true
          }
        },
      ]
      # Do not create an Virtual WAN resources
      vwan_hub_networks = []
    }

    # Set the default location
    location = var.primary_location
    # Create a custom tags input
    tags = var.connectivity_resources_tags
  }
}
```

modules/connectivity/variables.tf

The modules/connectivity/variables.tf file is used to declare a number of variables needed to configure this module. These are populated from the orchestration module, so no default values are specified.

```
# Use variables to customize the deployment

variable "root_id" {
  type        = string
  description = "Sets the value used for generating unique resource naming within the
module."
}

variable "primary_location" {
  type        = string
  description = "Sets the location for \"primary\" resources to be created in."
}

variable "secondary_location" {
  type        = string
  description = "Sets the location for \"secondary\" resources to be created in."
}

variable "subscription_id_connectivity" {
  type        = string
  description = "Subscription ID to use for \"connectivity\" resources."
}

variable "enable_ddos_protection" {
  type        = bool
  description = "Controls whether to create a DDoS Network Protection plan and link to
hub virtual networks."
}

variable "connectivity_resources_tags" {
  type        = map(string)
  description = "Specify tags to add to \"connectivity\" resources."
}
```

modules/core/lib/archetype_definition_customer_online.json

```
{
  "customer_online": {
    "policy_assignments": [
      "Deny-Resource-Locations",
      "Deny-RSG-Locations"
    ],
    "policy_definitions": [],
    "policy_set_definitions": [],
    "role_definitions": [],
    "archetype_config": {
      "parameters": {
        "Deny-Resource-Locations": {
          "listOfAllowedLocations": [
            "eastus",
            "eastus2",
            "westus",
            "northcentralus",
            "southcentralus"
          ]
        },
        "Deny-RSG-Locations": {
          "listOfAllowedLocations": [
            "eastus",
            "eastus2",
            "westus",
            "northcentralus",
            "southcentralus"
          ]
        }
      },
      "access_control": {}
    }
  }
}
```

modules/core/main.tf

The modules/core/main.tf file contains a customized module declaration to create the management group hierarchy and associated policies.

It also includes the necessary Terraform and provider configuration, and an azurerm_client_config resource which is used to determine the Tenant ID and Subscription ID values for the context being used to create these resources. This is used to ensure the deployment will target your Tenant Root Group by default, and to populate the subscription_id_xxxxx input variables.

```
# Configure Terraform to set the required AzureRM provider
# version and features{} block.

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

# Define the provider configuration

provider "azurerm" {
  features {}
}

# Get the current client configuration from the AzureRM provider.

data "azurerm_client_config" "current" {}

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

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

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

```
  # Base module configuration settings
  root_parent_id = data.azurerm_client_config.current.tenant_id
  root_id        = var.root_id
  root_name      = var.root_name
  library_path   = "${path.module}/lib"

  # Enable creation of the core management group hierarchy
  # and additional custom_landing_zones
  deploy_core_landing_zones = true
  custom_landing_zones      = local.custom_landing_zones

  # Configuration settings for identity resources is
  # bundled with core as no resources are actually created
  # for the identity subscription
  deploy_identity_resources    = true
  configure_identity_resources = local.configure_identity_resources
  subscription_id_identity     = var.subscription_id_identity

  # The following inputs ensure that managed parameters are
  # configured correctly for policies relating to connectivity
  # resources created by the connectivity module instance and
  # to map the subscription to the correct management group,
  # but no resources are created by this module instance
  deploy_connectivity_resources    = false
  configure_connectivity_resources = var.configure_connectivity_resources
  subscription_id_connectivity     = var.subscription_id_connectivity

  # The following inputs ensure that managed parameters are
  # configured correctly for policies relating to management
  # resources created by the management module instance and
  # to map the subscription to the correct management group,
  # but no resources are created by this module instance
  deploy_management_resources    = false
  configure_management_resources = var.configure_management_resources
  subscription_id_management     = var.subscription_id_management

}
```

modules/core/settings.core.tf

The modules/core/settings.core.tf file is used to specify the configuration used for creating the required core resources.

This is used as an input to the core module instance only, defining which additional management groups to create and to demonstrate some simple custom archetype configuration options.

```
# Configure the custom landing zones to deploy in
# addition to the core resource hierarchy
locals {
  custom_landing_zones = {
    "${var.root_id}-online-example-1" = {
      display_name               = "${upper(var.root_id)} Online Example 1"
      parent_management_group_id = "${var.root_id}-landing-zones"
      subscription_ids           = []
      archetype_config = {
        archetype_id   = "customer_online"
        parameters     = {}
        access_control = {}
      }
    }
    "${var.root_id}-online-example-2" = {
      display_name               = "${upper(var.root_id)} Online Example 2"
      parent_management_group_id = "${var.root_id}-landing-zones"
      subscription_ids           = []
      archetype_config = {
        archetype_id = "customer_online"
        parameters = {
          Deny-Resource-Locations = {
            listOfAllowedLocations = [
              var.primary_location,
              var.secondary_location,
            ]}
          Deny-RSG-Locations = {
            listOfAllowedLocations = [
              var.primary_location,
              var.secondary_location,
            ]
          }
        }
        access_control = {}
      }
    }
  }
}
```

modules/core/settings.identity.tf

The modules/core/settings.identity.tf file is used to specify the configuration used for configuring policies relating to the identity resources.

In this example we are setting the Deny-Subnet-Without-Nsg policy assignment enforcementMode to DoNotEnforce.

```
# Configure custom identity resources settings
locals {
  configure_identity_resources = {
    settings = {
      identity = {
        config = {
          # Disable this policy as can conflict with Terraform
          enable_deny_subnet_without_nsg = false
        }
      }
    }
  }
}
```

modules/core/variables.tf

The modules/core/variables.tf file is used to declare a number of variables needed to configure this module. These are populated from the orchestration module, so no default values are specified.

```
# Use variables to customize the deployment

variable "root_id" {
  type        = string
  description = "Sets the value used for generating unique resource naming within the
module."
}

variable "root_name" {
  type        = string
  description = "Sets the value used for the \"intermediate root\" management group
display name."
}

variable "primary_location" {
  type        = string
  description = "Sets the location for \"primary\" resources to be created in."
}
```

```
variable "secondary_location" {
  type        = string
  description = "Sets the location for \"secondary\" resources to be created in."
}

variable "subscription_id_connectivity" {
  type        = string
  description = "Subscription ID to use for \"connectivity\" resources."
}

variable "subscription_id_identity" {
  type        = string
  description = "Subscription ID to use for \"identity\" resources."
}

variable "subscription_id_management" {
  type        = string
  description = "Subscription ID to use for \"management\" resources."
}

variable "configure_connectivity_resources" {
  type        = any
  description = "Configuration settings for \"connectivity\" resources."
}

variable "configure_management_resources" {
  type        = any
  description = "Configuration settings for \"management\" resources."
}
```

modules/management/main.tf

The modules/management/main.tf file contains a customized module declaration to the Log Analytics workspace, Automation Account and Azure Monitor solutions in your management Subscription.

It also includes the necessary Terraform and provider configuration, and an azurerm_client_config resource which is used to determine the Tenant ID and Subscription ID values for the context being used to create these resources. This is used to ensure the deployment will target your Tenant Root Group by default, and to populate the subscription_id_management input variable.

```
# Configure Terraform to set the required AzureRM provider
# version and features{} block

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

# Define the provider configuration

provider "azurerm" {
  features {}

  subscription_id = var.subscription_id_management
}

# Get the current client configuration from the AzureRM provider

data "azurerm_client_config" "current" {}

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

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

  providers = {
    azurerm              = azurerm
    azurerm.connectivity = azurerm
    azurerm.management   = azurerm
```

```
  }

  # Base module configuration settings
  root_parent_id = data.azurerm_client_config.current.tenant_id
  root_id        = var.root_id

  # Disable creation of the core management group hierarchy
  # as this is being created by the core module instance
  deploy_core_landing_zones = false

  # Configuration settings for management resources
  deploy_management_resources    = true
  configure_management_resources = local.configure_management_resources
  subscription_id_management     = var.subscription_id_management

}
```

modules/management/outputs.tf

The modules/management/outputs.tf file contains modules outputs used when connecting the module instances together.

The configuration output is an important part of this example, as this is used to ensure the same values used to configure the management resources is shared with the core module instance. This ensures that managed parameters for policies deployed by the core module instance are configured with values correctly reflecting the resources deployed by this module instance.

```
# Output a copy of configure_management_resources for use
# by the core module instance

output "configuration" {
  description = "Configuration settings for the \"management\" resources."
  value       = local.configure_management_resources
}
```

modules/management/settings.management.tf

The modules/management/settings.management.tf file is used to specify the configuration used for creating the required management resources.

This is used as an input to the management module instance, but is also shared with the core module instance to ensure consistent configuration between resources and policies.

```
# Configure custom management resources settings
locals {
  configure_management_resources = {
    settings = {
      log_analytics = {
        config = {
          # Set a custom number of days to retain logs
          retention_in_days = var.log_retention_in_days
        }
      }
      security_center = {
        config = {
          # Configure a valid security contact email address
          email_security_contact = var.email_security_contact
        }
      }
    }
    # Set the default location
    location = var.primary_location
    # Create a custom tags input
    tags = var.management_resources_tags
  }
}
```

modules/management/variables.tf

The modules/management/variables.tf file is used to declare a number of variables needed to configure this module. These are populated from the orchestration module, so no default values are specified.

```
# Use variables to customize the deployment

variable "root_id" {
  type        = string
  description = "Sets the value used for generating unique resource naming within the
module."
}
```

```
variable "primary_location" {
  type        = string
  description = "Sets the location for \"primary\" resources to be created in."
}

variable "subscription_id_management" {
  type        = string
  description = "Subscription ID to use for \"management\" resources."
}

variable "email_security_contact" {
  type        = string
  description = "Set a custom value for the security contact email address."
}

variable "log_retention_in_days" {
  type        = number
  description = "Set a custom value for how many days to store logs in the Log Analytics
workspace."
}

variable "management_resources_tags" {
  type        = map(string)
  description = "Specify tags to add to \"management\" resources."
}
```

Deploy resources:

To simplify deployment, this example is deployed through a single root module.

From the root_module directory, simply run the following commands:

Ensure you have a connection correctly configured with permissions to Azure as per the Module permissions documentation
Initialize the Terraform workspace with the command terraform init
Generate a plan with the command terraform plan -out=tfplan
Review the output of the plan (use the command terraform show -json ./tfplan if you want to review the plan as a JSON file)
Start the deployment using the command terraform apply ./tfplan and follow the prompts
Once deployment is complete, review the created resources.

Number of Policies in This Deployment.

Count: 392

Number of Policy Assignment in This Deployment.

Count: 42

Number of Role Assignments in This Deployment.

Count: 24

Number of Resources in This Deployment.

Count: 84

Tree Structure:

```
rootAZURE-DEPLOYMENT# tree root_module/
root_module/
├── README.md
├── main.tf
├── modules
│   ├── connectivity
│   │   ├── main.tf
│   │   ├── output.tf
│   │   ├── settings.connectivity.tf
│   │   └── variables.tf
│   ├── core
│   │   ├── lib
│   │   │   └── archetype_definition_customer_online.json
│   │   ├── main.tf
│   │   ├── settings.core.tf
│   │   ├── settings.identity.tf
│   │   └── variables.tf
│   └── management
│       ├── main.tf
│       ├── output.tf
│       ├── settings.management.tf
│       └── variables.tf
└── variables.tf

5 directories, 16 files
rootAZURE-DEPLOYMENT#
```

Terraform Init:

```
- Finding hashicorp/time versions matching ">= 0.7.0"...
- Installing hashicorp/random v3.5.1...
- Installed hashicorp/random v3.5.1 (signed by HashiCorp)
- Installing azure/azapi v1.6.0...
- Installed azure/azapi v1.6.0 (signed by a HashiCorp partner, key ID 6F0B91BDE98478CF)
- Installing hashicorp/azurerm v3.56.0...
- Installed hashicorp/azurerm v3.56.0 (signed by HashiCorp)
- Installing hashicorp/time v0.9.1...
- Installed hashicorp/time v0.9.1 (signed by HashiCorp)

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 [ ~/ALZ-400/ALZ-400-2nd ]$
```

Terraform Plan:

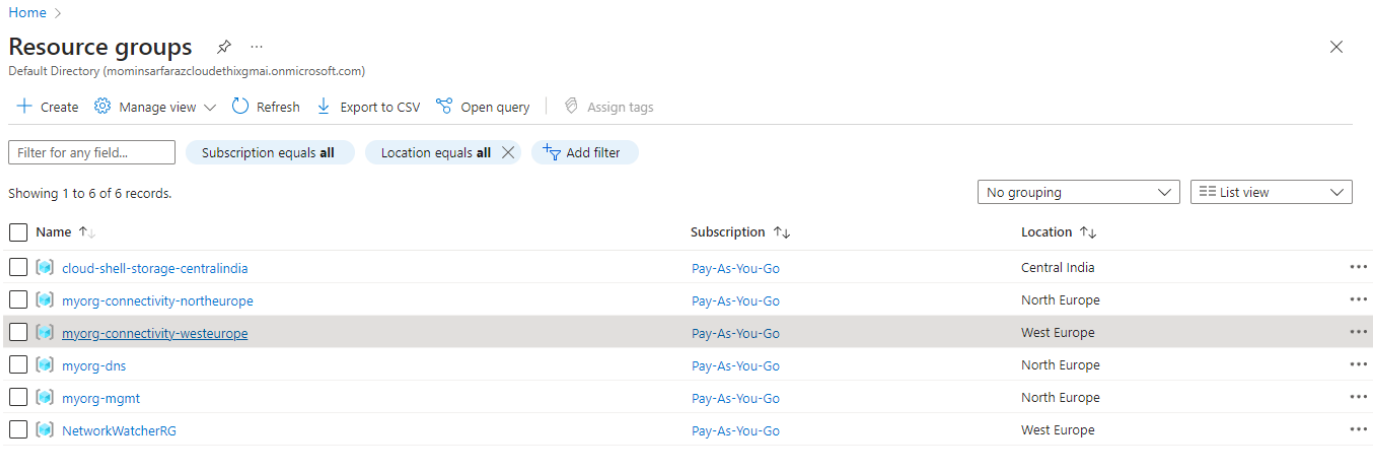
```
  # module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg/providers/Microsoft.Authorization/policyAssignments/Enforce-ACSB"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg/providers/Microsoft.Authorization/roleAssignments/20b87dbc-b70-5379-ad61-97a3ccecc927"] 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: 467 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.
```

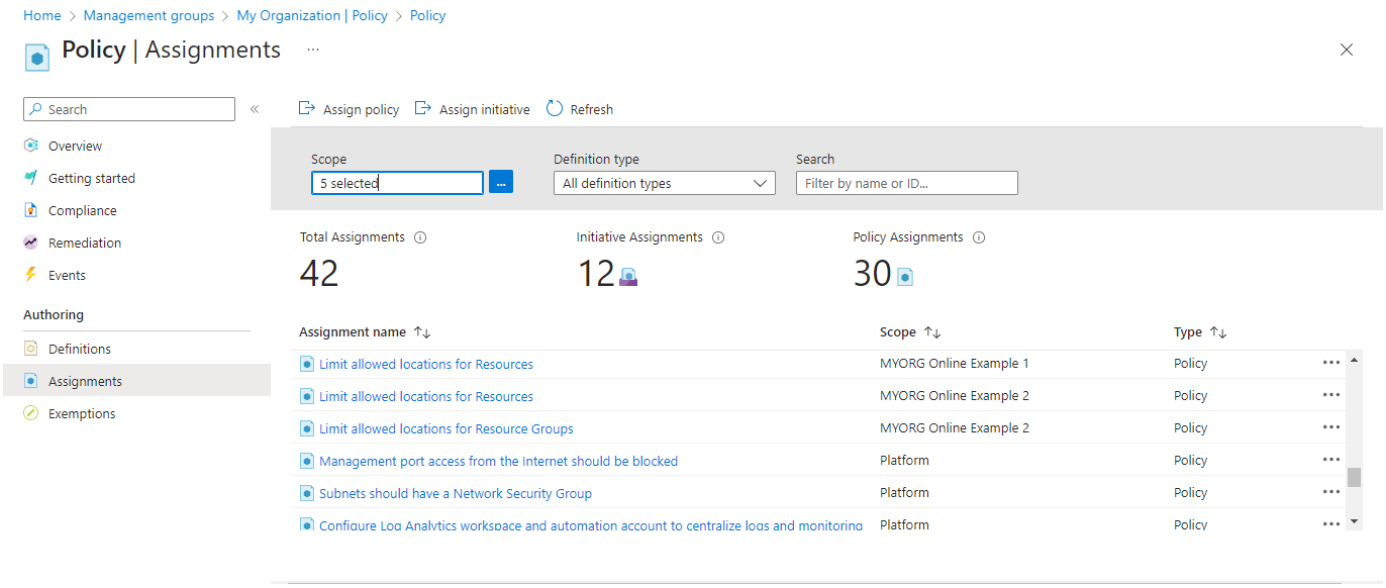
Terraform Apply:

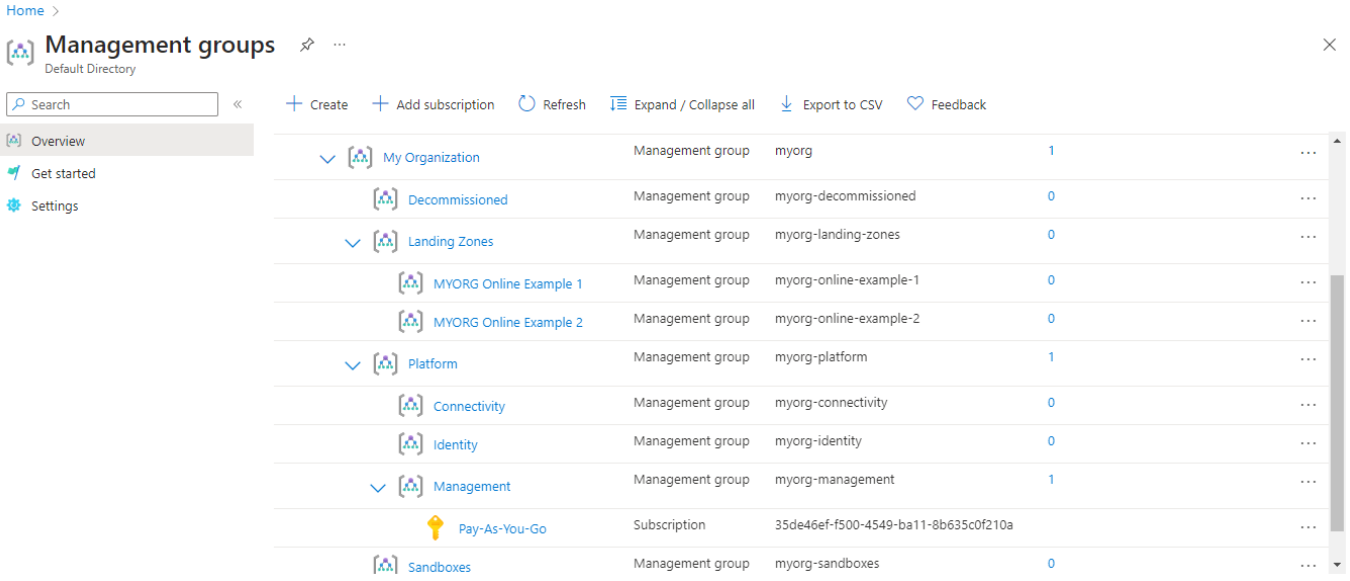
```
module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/policyAssignments/Deploy-AzSqlDb-Auditing"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/roleAssignments/6b33d798-88c0-5983-ae6e-ca12dd066b27"]: Still creating... [10s elapsed]
module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/policyAssignments/Deploy-AKS-Policy"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/roleAssignments/225d88be-c156-56d7-b3d8-98c92f89df55"]: Still creating... [20s elapsed]
module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-platform/providers/Microsoft.Authorization/policyAssignments/Deploy-VM-Backup"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-platform/providers/Microsoft.Authorization/roleAssignments/305b3c37-969e-52f7-b233-f507f98e06af"]: Still creating... [20s elapsed]
module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/policyAssignments/Deploy-AzSqlDb-Auditing"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/roleAssignments/6b33d798-88c0-5983-ae6e-ca12dd066b27"]: Still creating... [20s elapsed]
module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-platform/providers/Microsoft.Authorization/policyAssignments/Deploy-VM-Backup"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-platform/providers/Microsoft.Authorization/roleAssignments/305b3c37-969e-52f7-b233-f507f98e06af"]: Creation complete after 23s [id=/providers/Microsoft.Management/managementGroups/myorg-platform/providers/Microsoft.Authorization/roleAssignments/305b3c37-969e-52f7-b233-f507f98e06af]
module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/policyAssignments/Deploy-AzSqlDb-Auditing"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/roleAssignments/6b33d798-88c0-5983-ae6e-ca12dd066b27"]: Creation complete after 24s [id=/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/roleAssignments/6b33d798-88c0-5983-ae6e-ca12dd066b27]
module.core.module.alz.module.role_assignments_for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/policyAssignments/Deploy-AKS-Policy"].azurerm_role_assignment.for_policy["/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/roleAssignments/225d88be-c156-56d7-b3d8-98c92f89df55"]: Creation complete after 25s [id=/providers/Microsoft.Management/managementGroups/myorg-landing-zones/providers/Microsoft.Authorization/roleAssignments/225d88be-c156-56d7-b3d8-98c92f89df55]
module.core.module.alz.time_sleep.after_azurerm_role_assignment: Creating...
module.core.module.alz.time_sleep.after_azurerm_role_assignment: Creation complete after 0s [id=2023-05-18T09:12:14Z]

Apply complete! Resources: 467 added, 0 changed, 0 destroyed.
sarfaraz [ ~/ALZ-400/ALZ-400-2nd ]$
```

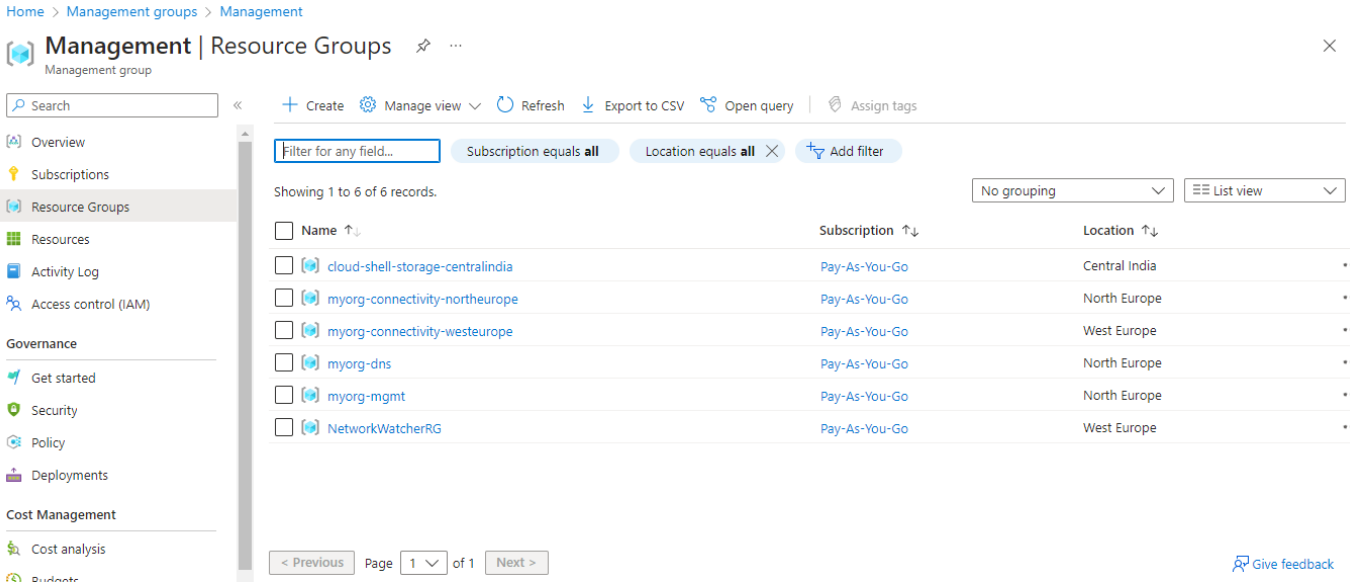
Terraform Destroy:

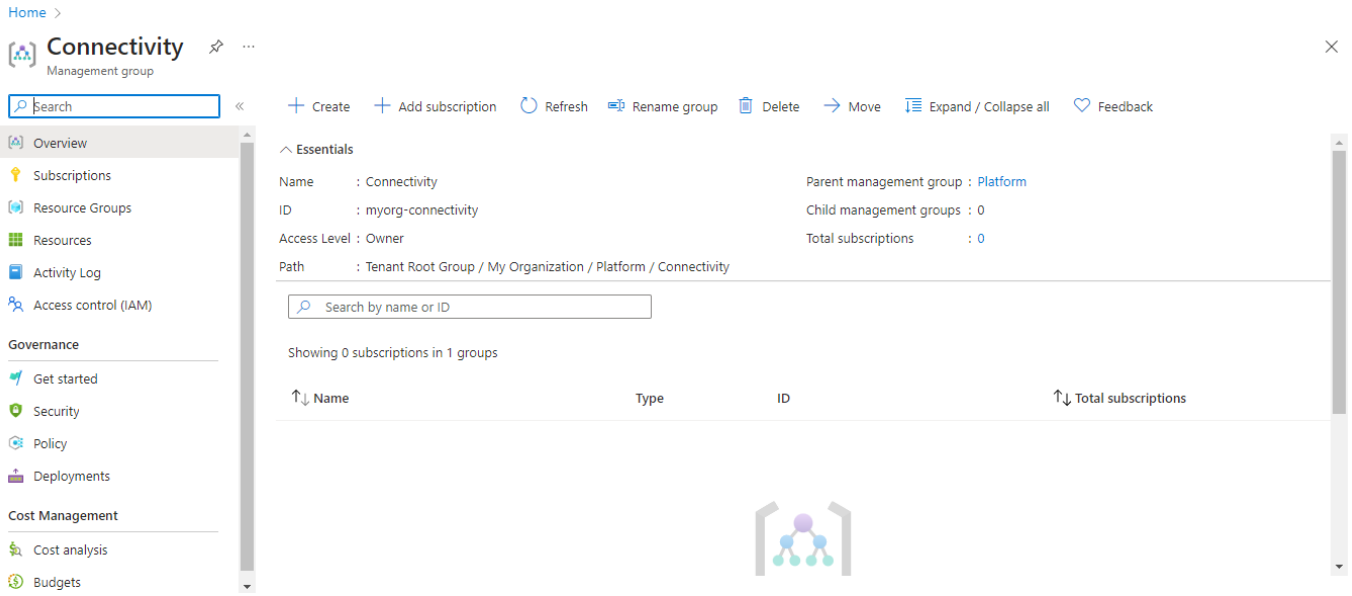
--SCREEN'S--


Home
Resource groups
Default Directory (mominsarfarazcloudethixgmai.onmicrosoft.com)
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Filter for any field...
Subscription equals all
Location equals all
Add filter
Showing 1 to 6 of 6 records.
No grouping
List view
Name
Subscription
Location
cloud-shell-storage-centralindia
Pay-As-You-Go
Central India
myorg-connectivity-northeurope
Pay-As-You-Go
North Europe
myorg-connectivity-westeurope
Pay-As-You-Go
West Europe
myorg-dns
Pay-As-You-Go
North Europe
myorg-mgmt
Pay-As-You-Go
North Europe
NetworkWatcherRG
Pay-As-You-Go
West Europe


Home
Management groups
My Organization | Policy
Policy
Policy | Assignments
Search
Overview
Getting started
Compliance
Remediation
Events
Authoring
Definitions
Assignments
Exemptions
Assign policy
Assign initiative
Refresh
Scope
5 selected
Definition type
All definition types
Search
Filter by name or ID...
Total Assignments
42
Initiative Assignments
12
Policy Assignments
30
Assignment name
Scope
Type
Limit allowed locations for Resources
MYORG Online Example 1
Policy
Limit allowed locations for Resources
MYORG Online Example 2
Policy
Limit allowed locations for Resource Groups
MYORG Online Example 2
Policy
Management port access from the Internet should be blocked
Platform
Policy
Subnets should have a Network Security Group
Platform
Policy
Configure Log Analytics workspace and automation account to centralize logs and monitoring
Platform
Policy


Home
Management groups
Default Directory
Search
Create
Add subscription
Refresh
Expand / Collapse all
Export to CSV
Feedback
Overview
Get started
Settings
My Organization Management group myorg 1
Decommissioned Management group myorg-decommissioned 0
Landing Zones Management group myorg-landing-zones 0
MYORG Online Example 1 Management group myorg-online-example-1 0
MYORG Online Example 2 Management group myorg-online-example-2 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


Home > Management groups > Management
Management | Resource Groups
Management group
Search
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Overview
Subscriptions
Resource Groups
Resources
Activity Log
Access control (IAM)
Governance
Get started
Security
Policy
Deployments
Cost Management
Cost analysis
Filter for any field...
Subscription equals all
Location equals all
Add filter
Showing 1 to 6 of 6 records.
No grouping
List view
Name
Subscription
Location
cloud-shell-storage-centralindia Pay-As-You-Go Central India
myorg-connectivity-northeurope Pay-As-You-Go North Europe
myorg-connectivity-westeurope Pay-As-You-Go West Europe
myorg-dns Pay-As-You-Go North Europe
myorg-mgmt Pay-As-You-Go North Europe
NetworkWatcherRG Pay-As-You-Go West Europe
< Previous
Page 1 of 1
Next >
Give feedback


Home
Connectivity
Management group
Search
Overview
Subscriptions
Resource Groups
Resources
Activity Log
Access control (IAM)
Governance
Get started
Security
Policy
Deployments
Cost Management
Cost analysis
Budgets
Create
Add subscription
Refresh
Rename group
Delete
Move
Expand / Collapse all
Feedback
Essentials
Name : Connectivity
ID : myorg-connectivity
Access Level : Owner
Path : Tenant Root Group / My Organization / Platform / Connectivity
Parent management group : Platform
Child management groups : 0
Total subscriptions : 0
Search by name or ID
Showing 0 subscriptions in 1 groups
Name
Type
ID
Total subscriptions

--END--