# Azure ALZ - 400

## (GITHUB)

## Deploy using multiple module declarations with remote state

Terraform Resoruce Count: 519
Connectivity: 76
Management: 217
Core: 226
Total Policy Count:  391

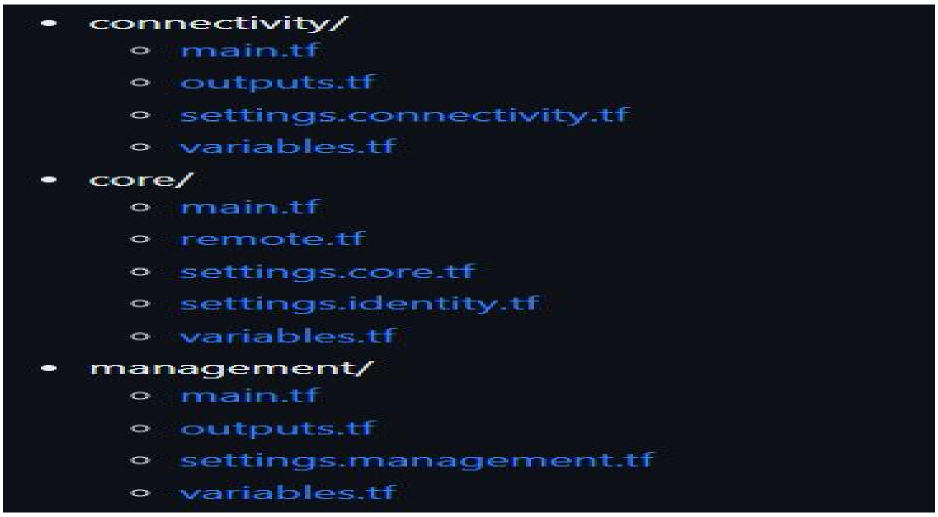
**Scope**: Deploying Azure landing zone using multiple declarations of the module using remote state to support running in multiple Terraform workspace.

**Customization:** The main motive of this deployment is to have the deployment with multiple tf state files and to control the configuration of resources by scope. For this, the code was split into the below parts.

| Resource FIles | Connectivity | Core | Management |
|---|---|---|---|
| Main.tf | ☑ | ☑ | ☑ |
| Outputs.tf | ☑ | ☑ | ☑ |
| Settings.tf | ☑<br>(settings.connectivity.tf) | ☑<br>(settings.core.tf, settings.identity.tf) | ☑<br>(settings.management.tf) |
| Variables.tf | ☑ | ☑ | ☑ |
| Archetypes.json | ☒ | ☑ | ☒ |

As shown above, there is **no root module** to control these sub-modules. So, they need to be deployed as separate and hence, would have different state files.

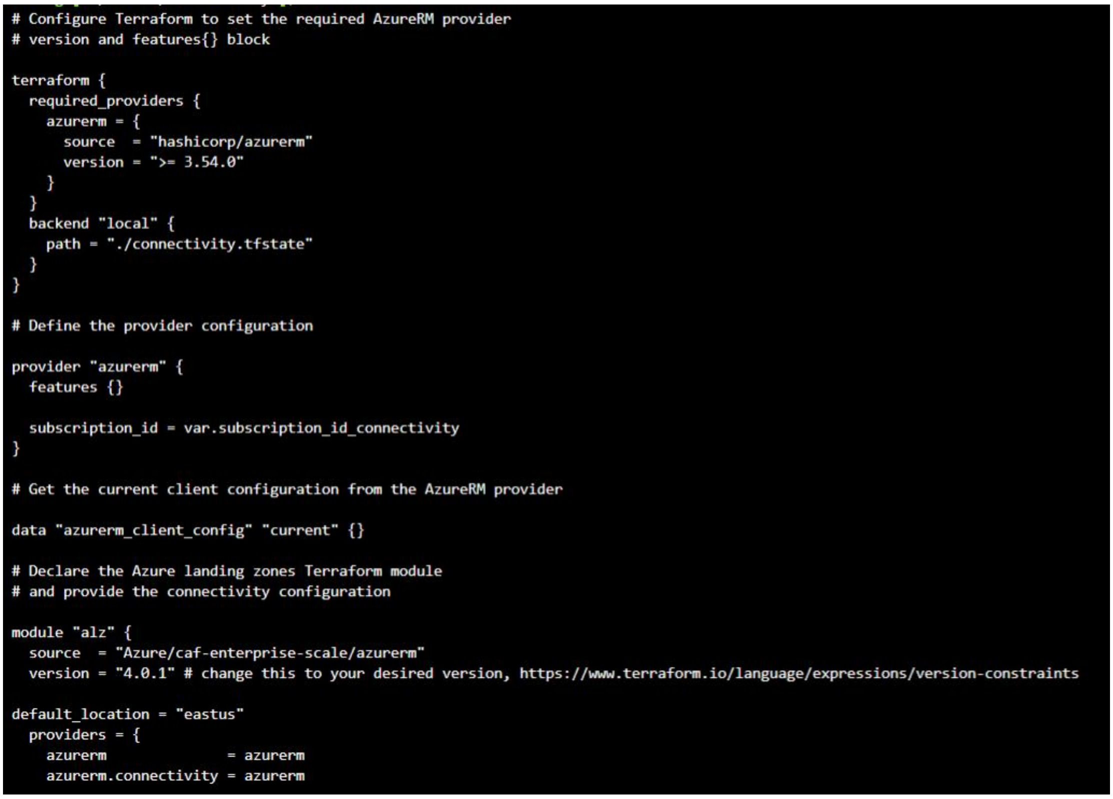
## Below is the tree structure of the code.



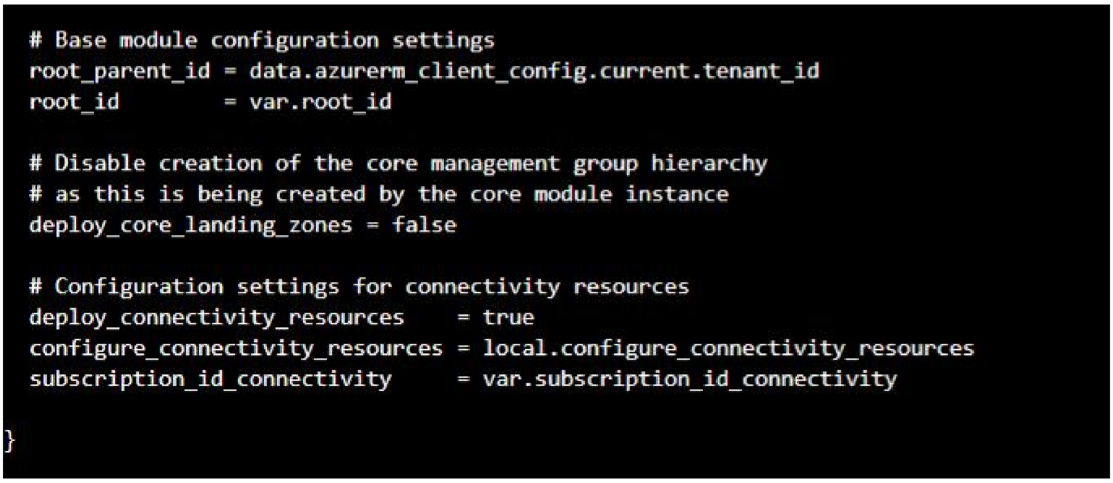
- connectivity/
  - main.tf
  - outputs.tf
  - settings.connectivity.tf
  - variables.tf
- core/
  - main.tf
  - remote.tf
  - settings.core.tf
  - settings.identity.tf
  - variables.tf
- management/
  - main.tf
  - outputs.tf
  - settings.management.tf
  - variables.tf

## Connectivity

Main.tf

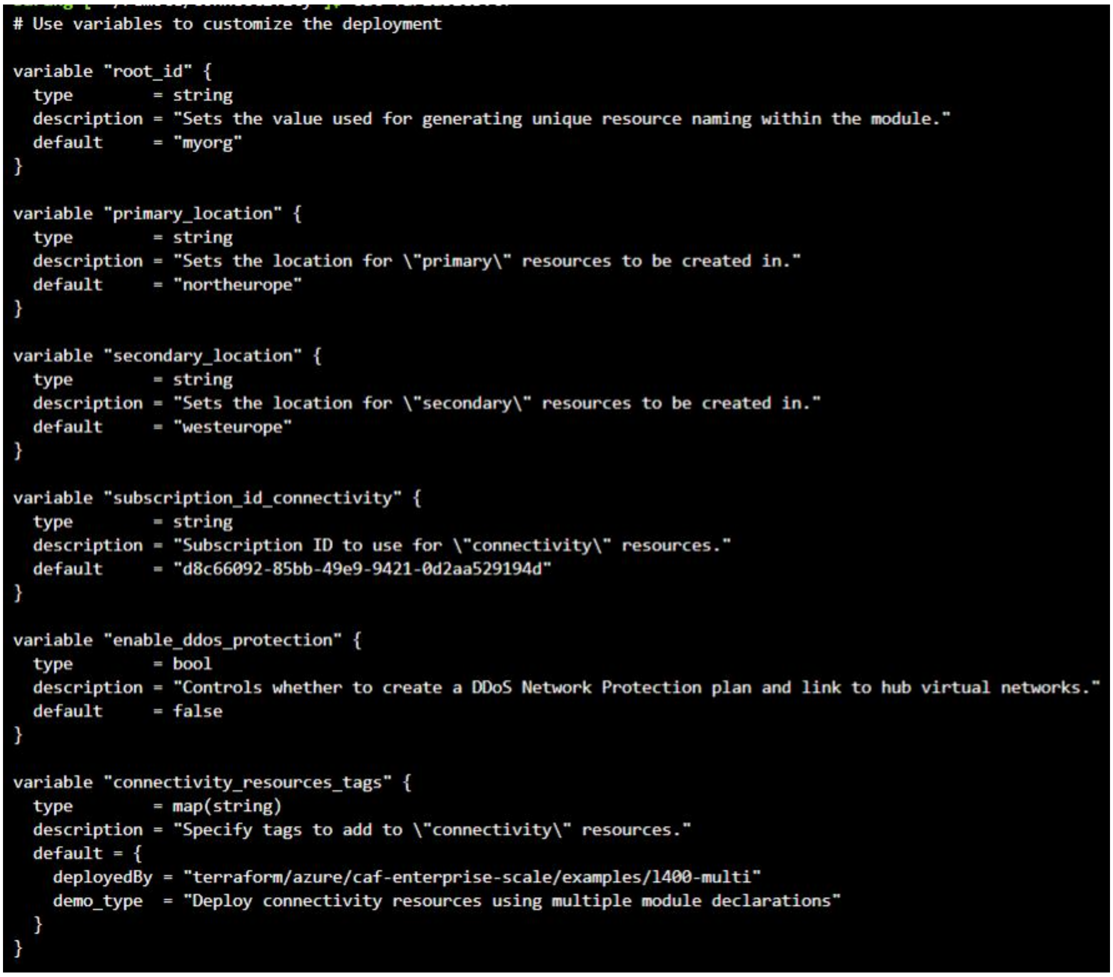


```
# Configure Terraform to set the required AzureRM provider
# version and features{} block

terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.54.0"
    }
  }
  backend "local" {
    path = "./connectivity.tfstate"
  }
}

# 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 = "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
```



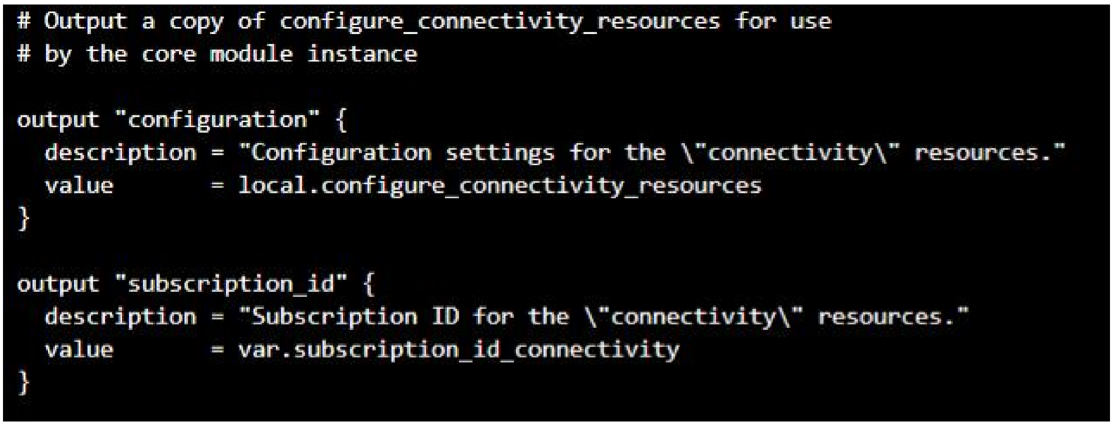
```
  # Base module configuration settings
  root_parent_id = data.azurerm_client_config.current.tenant_id
  root_id        = var.root_id

  # 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

}
```

Variables.tf



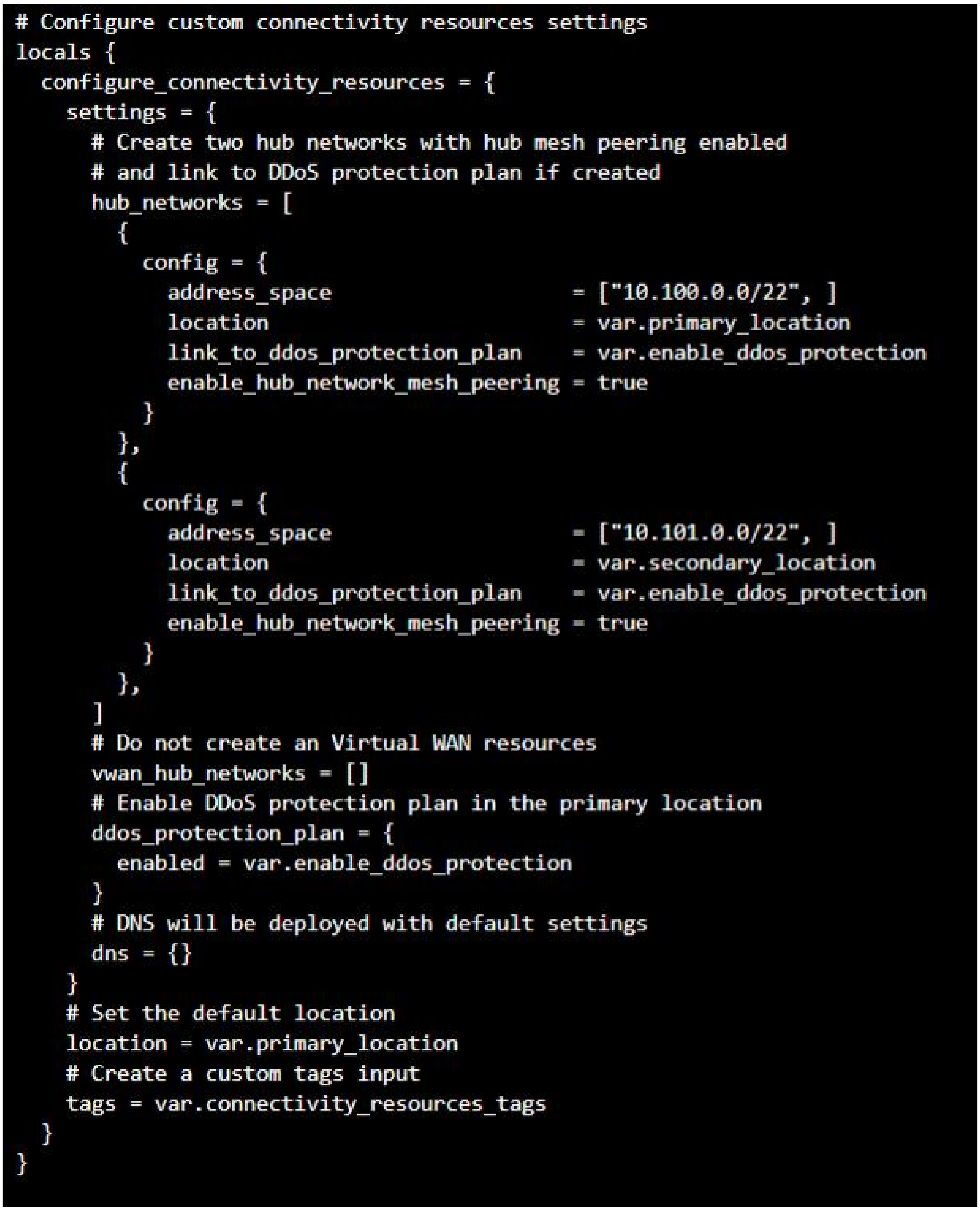
```
# Use variables to customize the deployment

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

variable "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     = "d8c66092-85bb-49e9-9421-0d2aa529194d"
}

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"
  }
}
```

Outputs.tf



```
# 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
}

output "subscription_id" {
  description = "Subscription ID for the \"connectivity\" resources."
  value       = var.subscription_id_connectivity
}
```

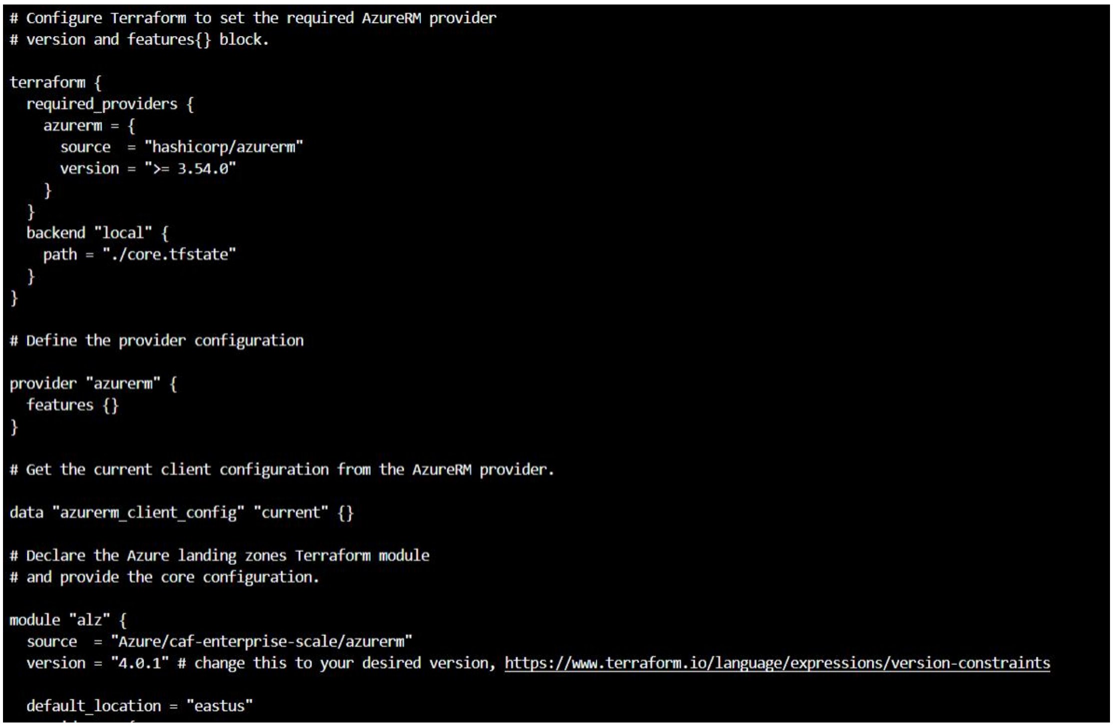
Settings.connectivity.tf



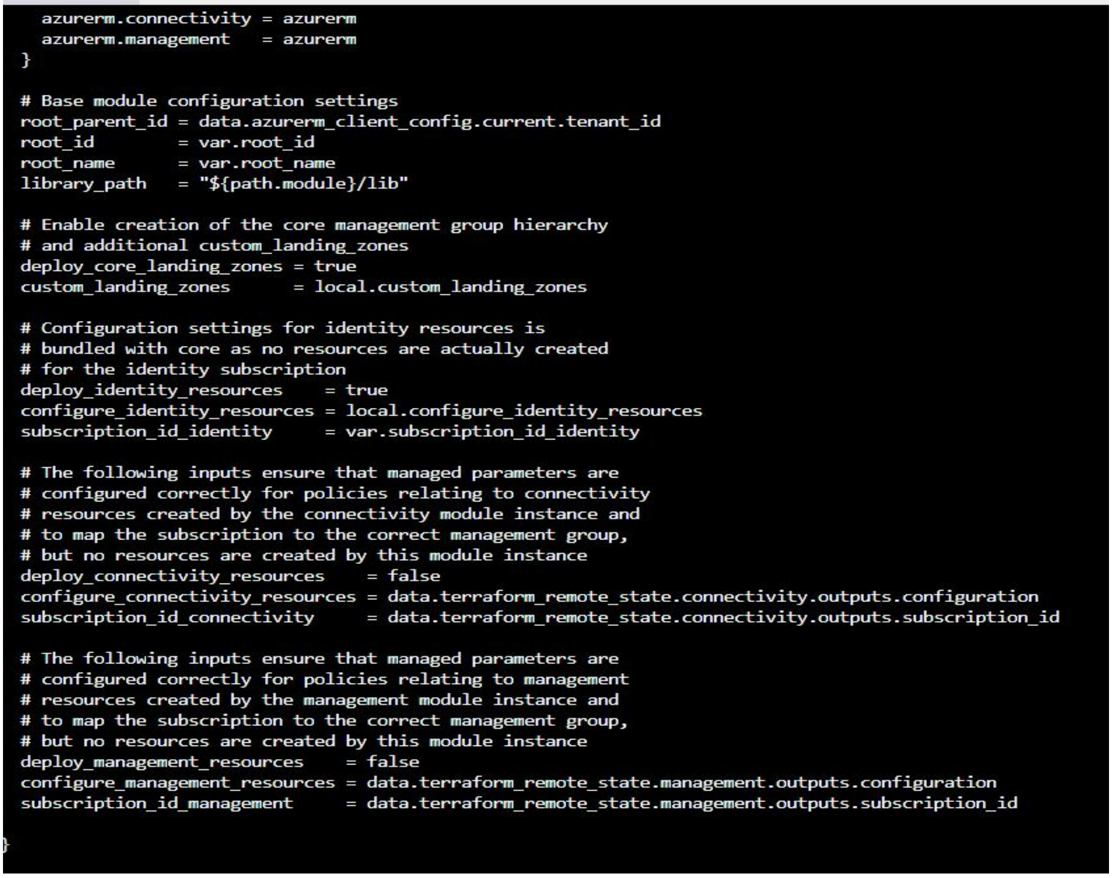
```
# Configure custom connectivity resources settings
locals {
  configure_connectivity_resources = {
    settings = {
      # Create two hub networks with hub mesh peering enabled
      # and link to DDoS protection plan if created
      hub_networks = [
        {
          config = {
            address_space                   = ["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 = []
      # Enable DDoS protection plan in the primary location
      ddos_protection_plan = {
        enabled = var.enable_ddos_protection
      }
      # DNS will be deployed with default settings
      dns = {}
    }
    # Set the default location
    location = var.primary_location
    # Create a custom tags input
    tags = var.connectivity_resources_tags
  }
}
```

# Core

## Main.tf

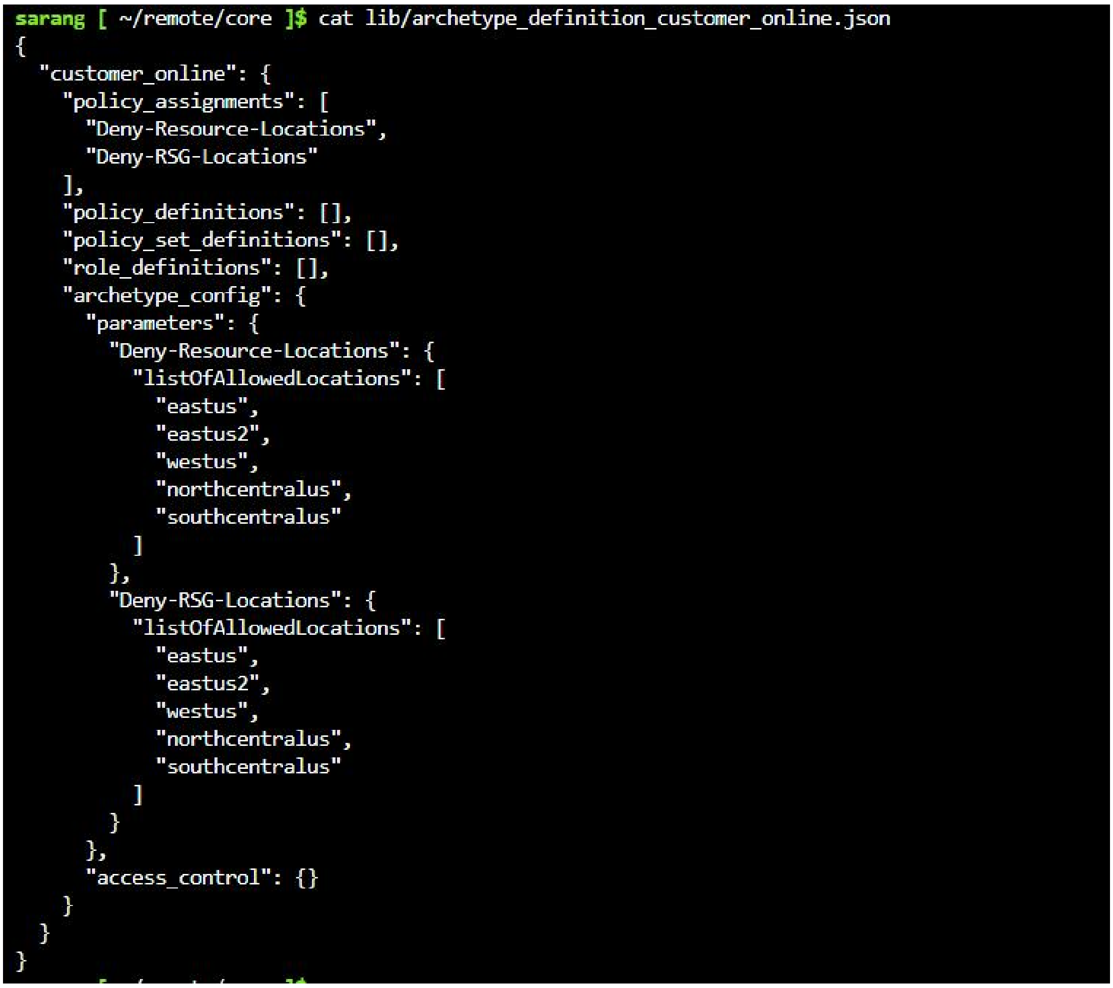


```
# Configure Terraform to set the required AzureRM provider
# version and features{} block.

terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.54.0"
    }
  }
  backend "local" {
    path = "./core.tfstate"
  }
}

# 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

  default_location = "eastus"
```



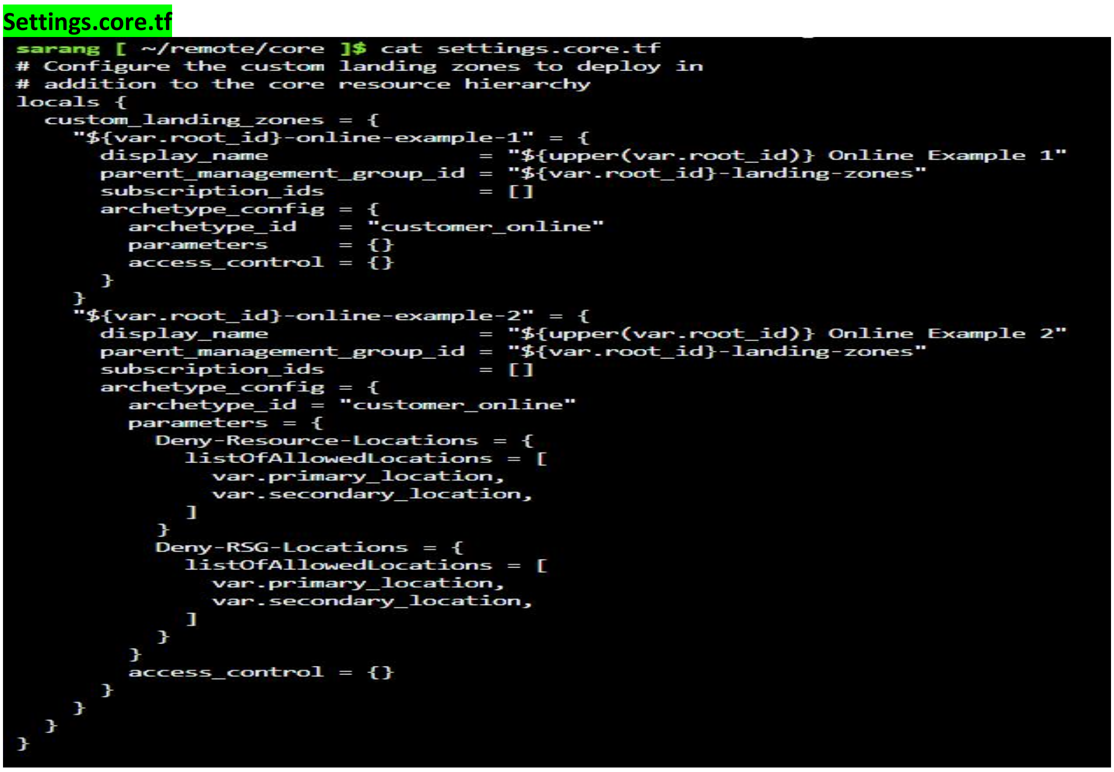
```
   azurerm.connectivity = azurerm
   azurerm.management   = azurerm
 }

 # 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 = data.terraform_remote_state.connectivity.outputs.configuration
 subscription_id_connectivity     = data.terraform_remote_state.connectivity.outputs.subscription_id

 # 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 = data.terraform_remote_state.management.outputs.configuration
 subscription_id_management     = data.terraform_remote_state.management.outputs.subscription_id

}
```

lib/archetype_definition_customer_online.json



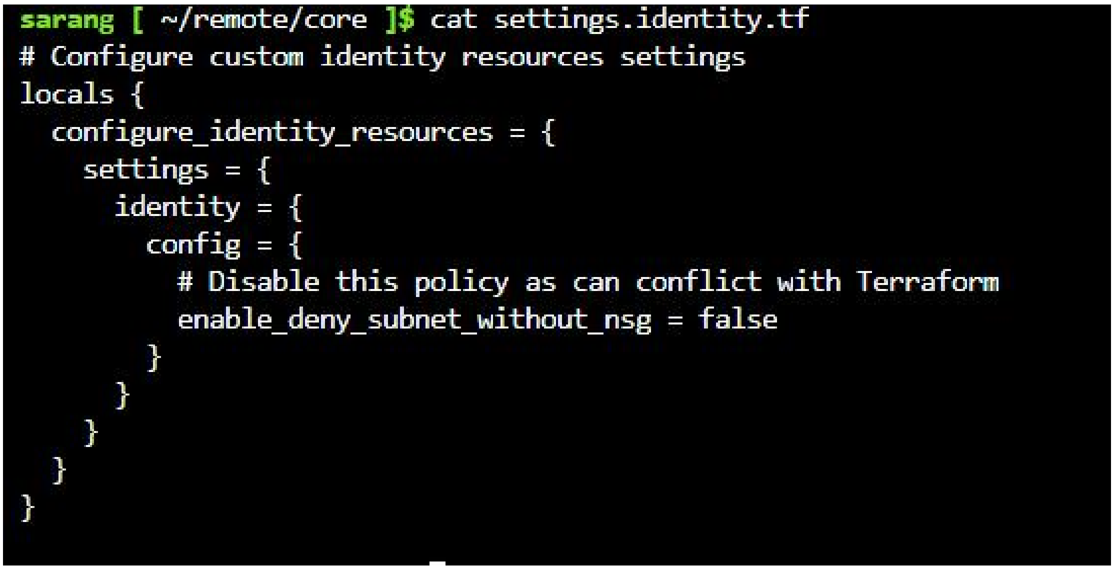
```
sarang [ ~/remote/core ]$ cat lib/archetype_definition_customer_online.json
{
  "customer_online": {
    "policy_assignments": [
      "Deny-Resource-Locations",
      "Deny-RSG-Locations"
    ],
    "policy_definitions": [],
    "policy_set_definitions": [],
    "role_definitions": [],
    "archetype_config": {
      "parameters": {
        "Deny-Resource-Locations": {
          "listOfAllowedLocations": [
            "eastus",
            "eastus2",
            "westus",
            "northcentralus",
            "southcentralus"
          ]
        },
        "Deny-RSG-Locations": {
          "listOfAllowedLocations": [
            "eastus",
            "eastus2",
            "westus",
            "northcentralus",
            "southcentralus"
          ]
        }
      },
      "access_control": {}
    }
  }
}
```

Settings.core.tf



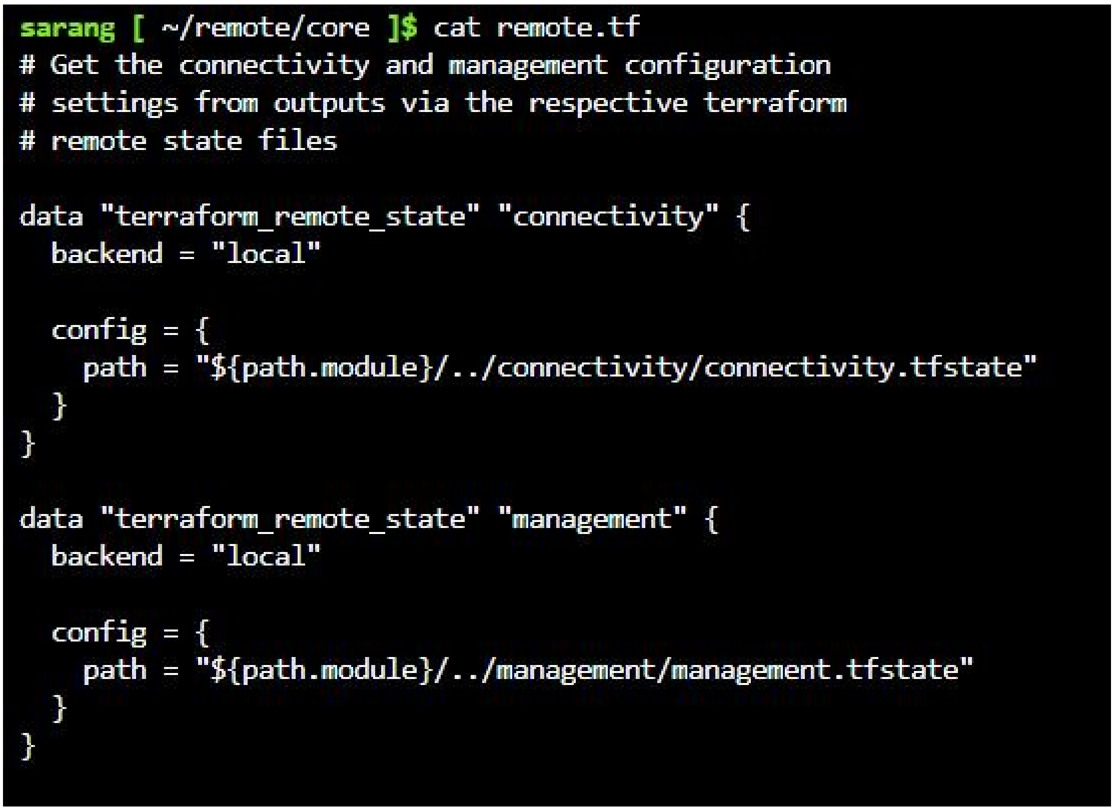
```
sarang [ ~/remote/core ]$ cat settings.core.tf
# Configure the custom landing zones to deploy in
# addition to the core resource hierarchy
locals {
  custom_landing_zones = {
    "${var.root_id}-online-example-1" = {
      display_name               = "${upper(var.root_id)} Online Example 1"
      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 = {}
      }
    }
  }
}
```

Settings.identity.tf



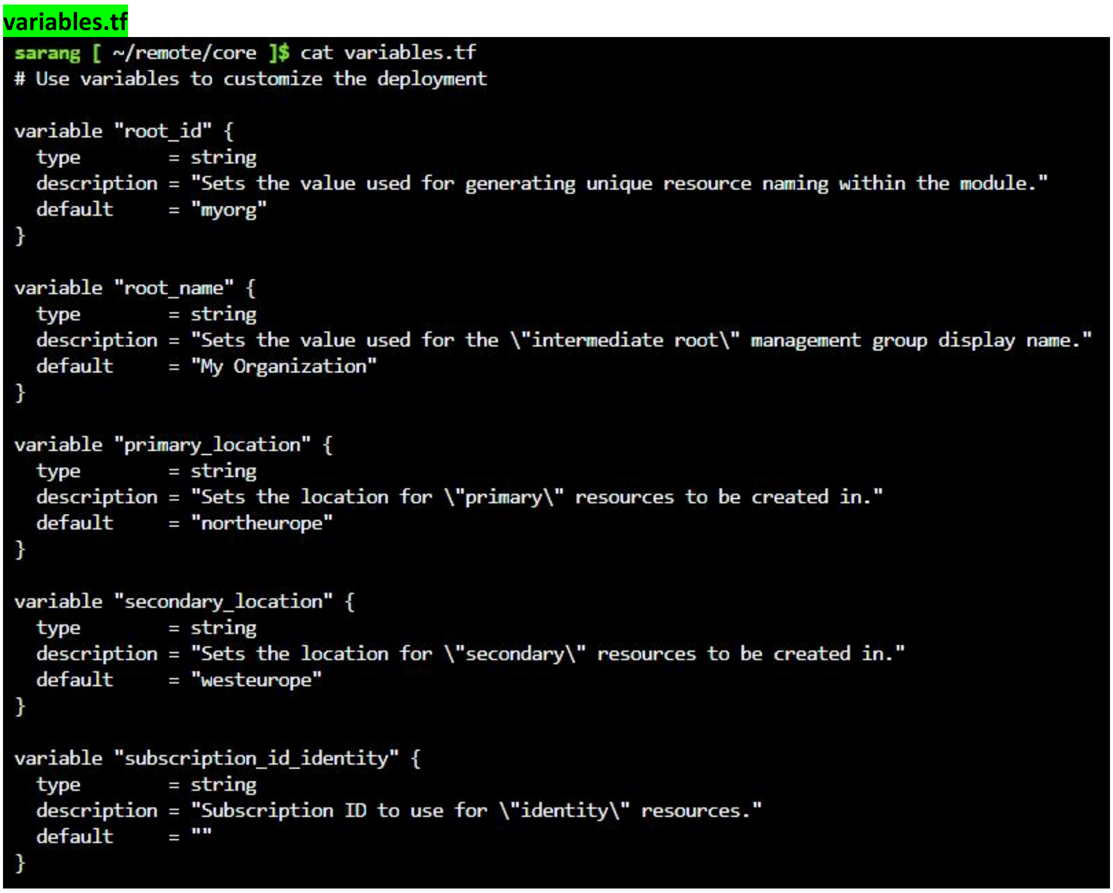
```
sarang [ ~/remote/core ]$ cat settings.identity.tf
# Configure custom identity resources settings
locals {
  configure_identity_resources = {
    settings = {
      identity = {
        config = {
          # Disable this policy as can conflict with Terraform
          enable_deny_subnet_without_nsg = false
        }
      }
    }
  }
}
```

remote.tf



```
sarang [ ~/remote/core ]$ cat remote.tf
# Get the connectivity and management configuration
# settings from outputs via the respective terraform
# remote state files

data "terraform_remote_state" "connectivity" {
  backend = "local"

  config = {
    path = "${path.module}/../connectivity/connectivity.tfstate"
  }
}

data "terraform_remote_state" "management" {
  backend = "local"

  config = {
    path = "${path.module}/../management/management.tfstate"
  }
}
```

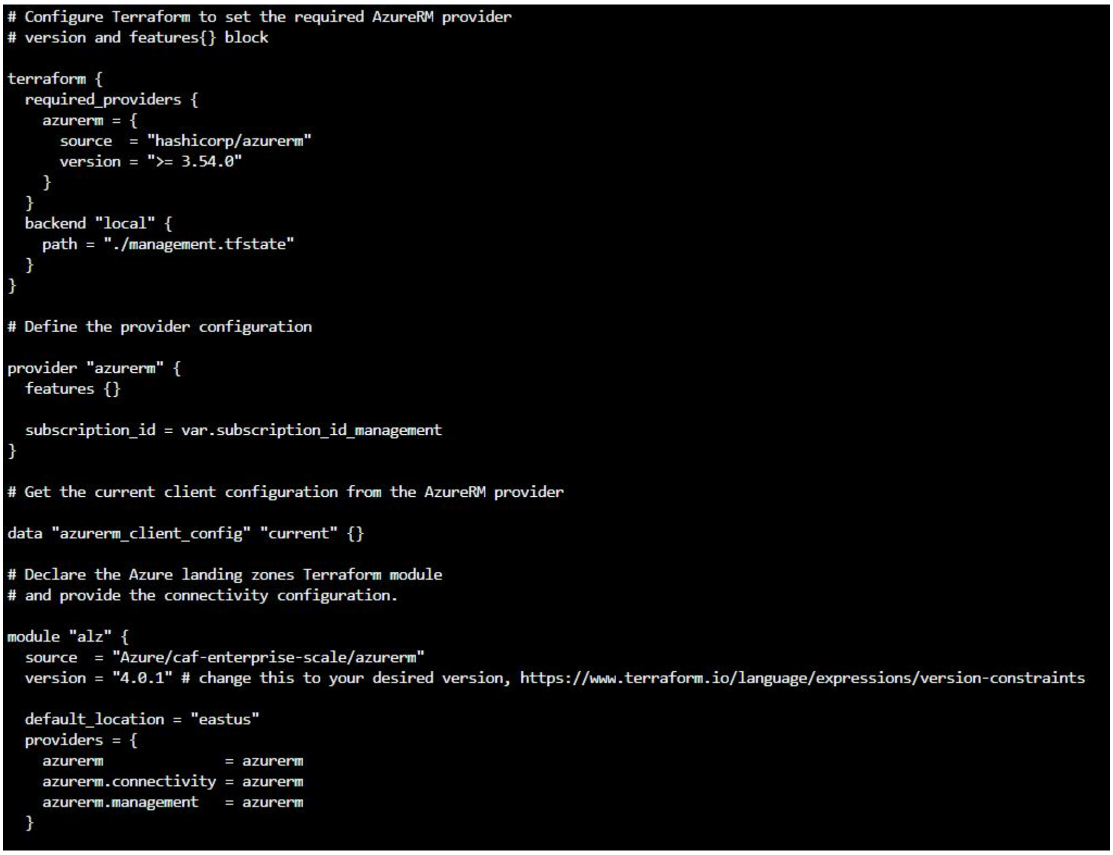
variables.tf



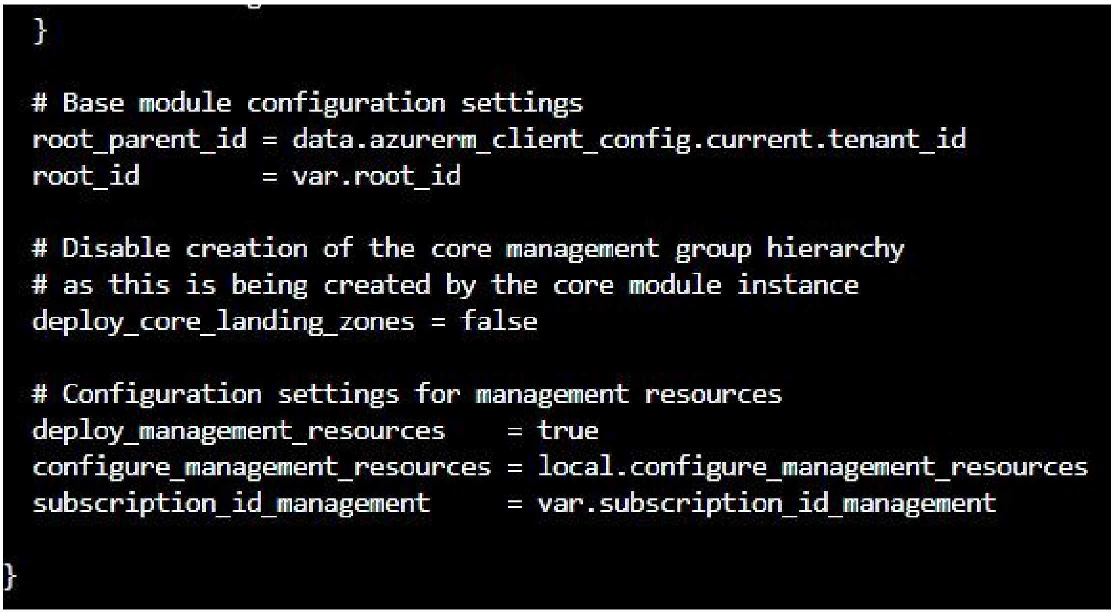
```
sarang [ ~/remote/core ]$ cat variables.tf
# Use variables to customize the deployment

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

# Management

Main.tf

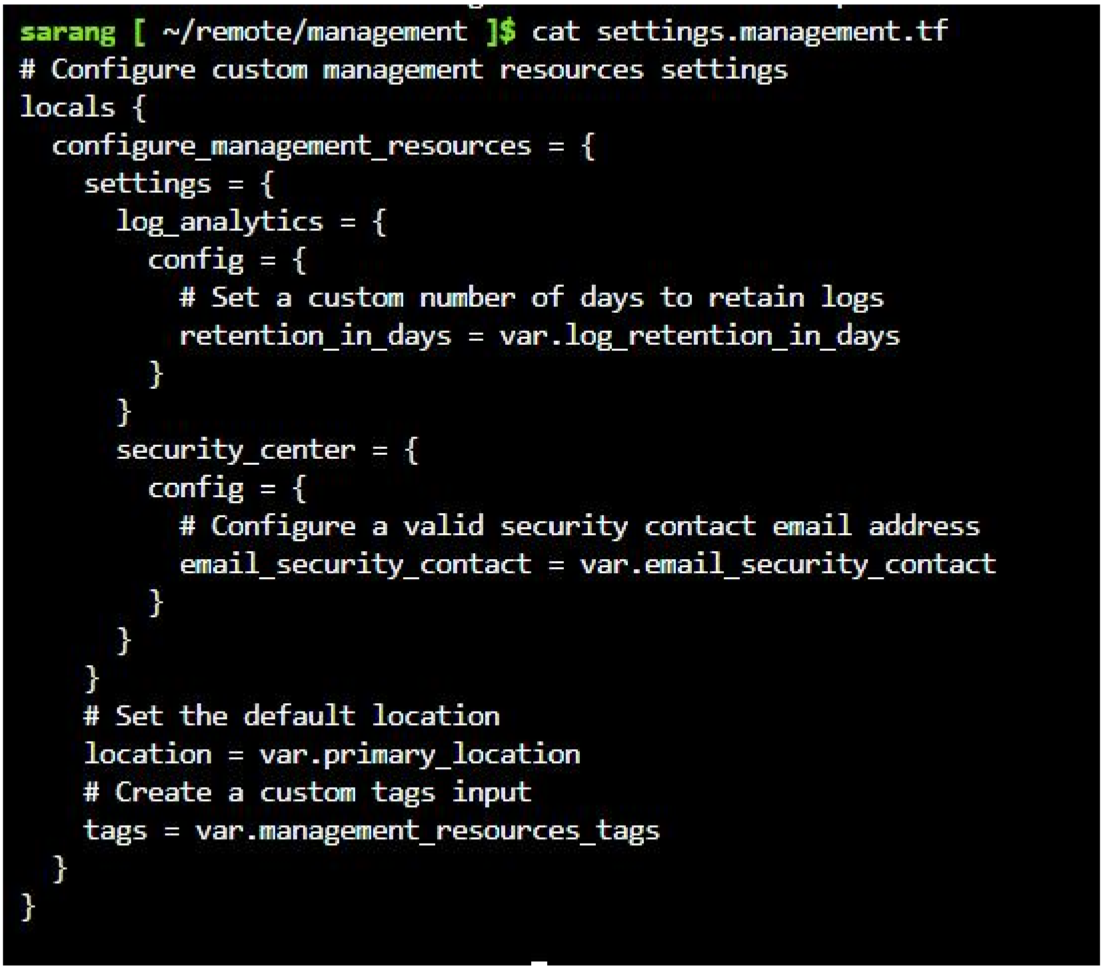


```
# Configure Terraform to set the required AzureRM provider
# version and features{} block

terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.54.0"
    }
  }
  backend "local" {
    path = "./management.tfstate"
  }
}

# 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

  default_location = "eastus"
  providers = {
    azurerm              = azurerm
    azurerm.connectivity = azurerm
    azurerm.management   = azurerm
  }
```



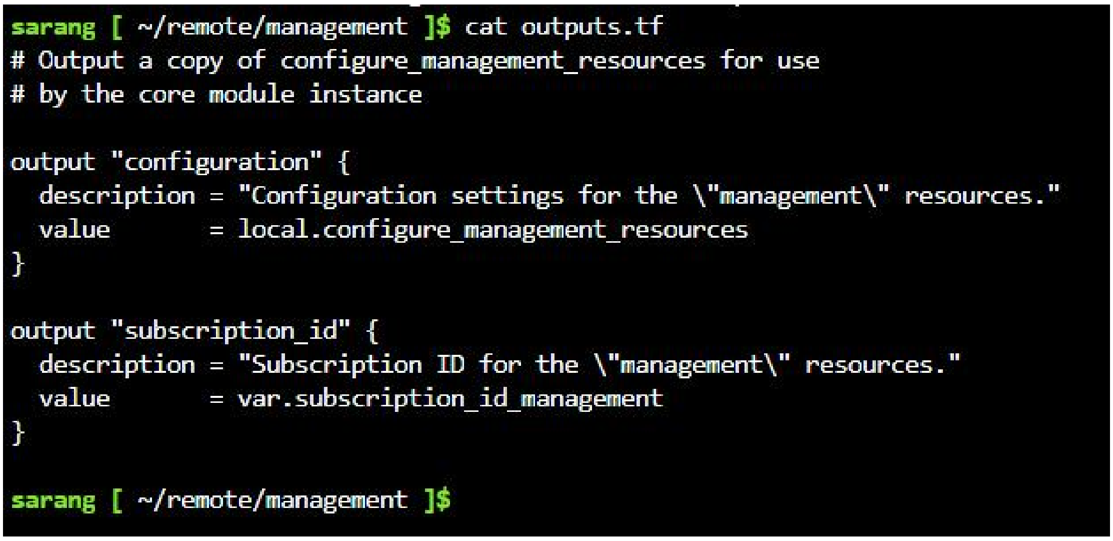
```
  }

  # 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 management resources
  deploy_management_resources    = true
  configure_management_resources = local.configure_management_resources
  subscription_id_management     = var.subscription_id_management

}
```

Settings.management.tf



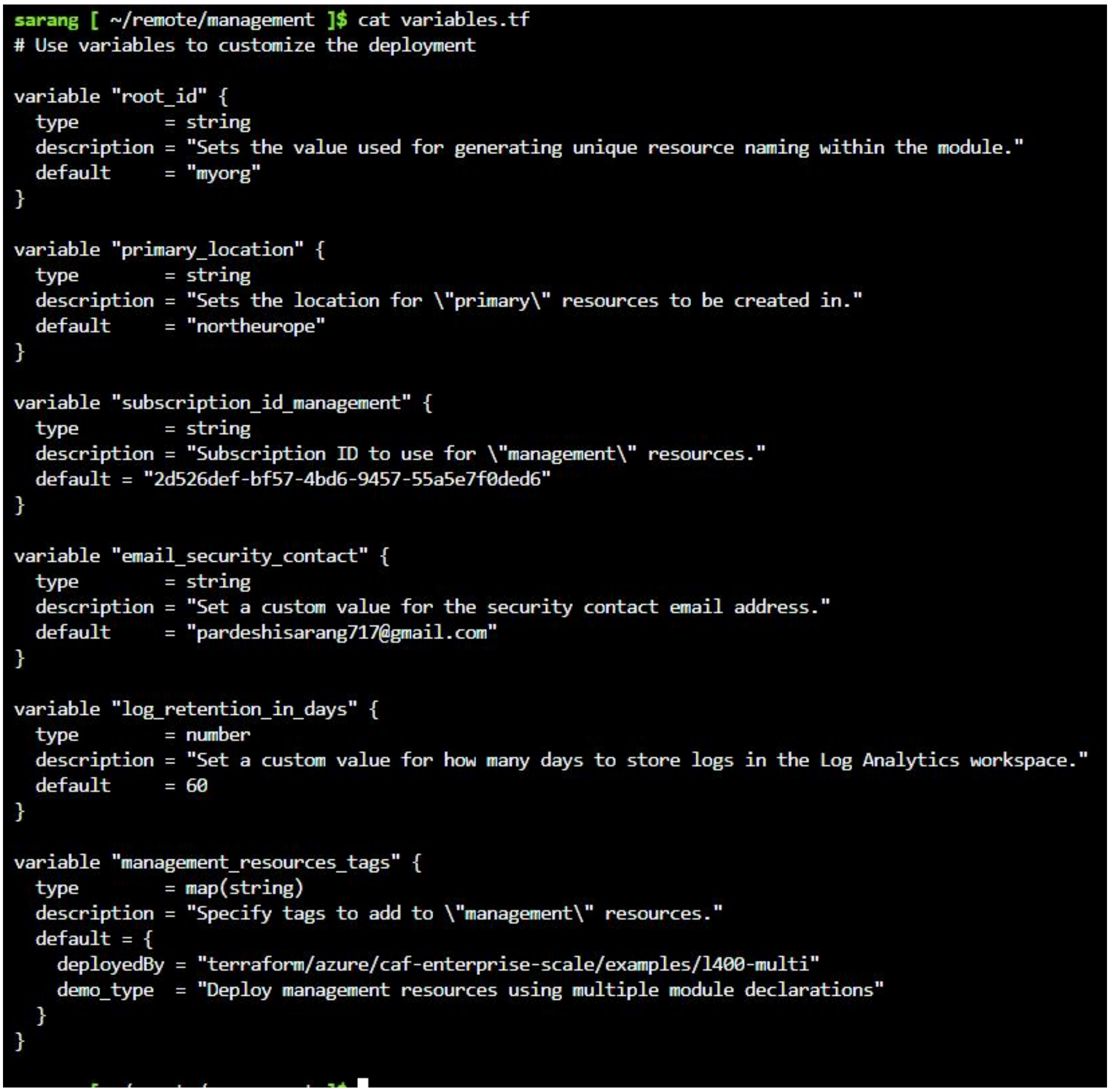
```
sarang [ ~/remote/management ]$ cat settings.management.tf
# Configure custom management resources settings
locals {
  configure_management_resources = {
    settings = {
      log_analytics = {
        config = {
          # Set a custom number of days to retain logs
          retention_in_days = 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
  }
}
```

outputs.tf



```
sarang [ ~/remote/management ]$ cat outputs.tf
# 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
}

output "subscription_id" {
  description = "Subscription ID for the \"management\" resources."
  value       = var.subscription_id_management
}

sarang [ ~/remote/management ]$
```

Variables.tf

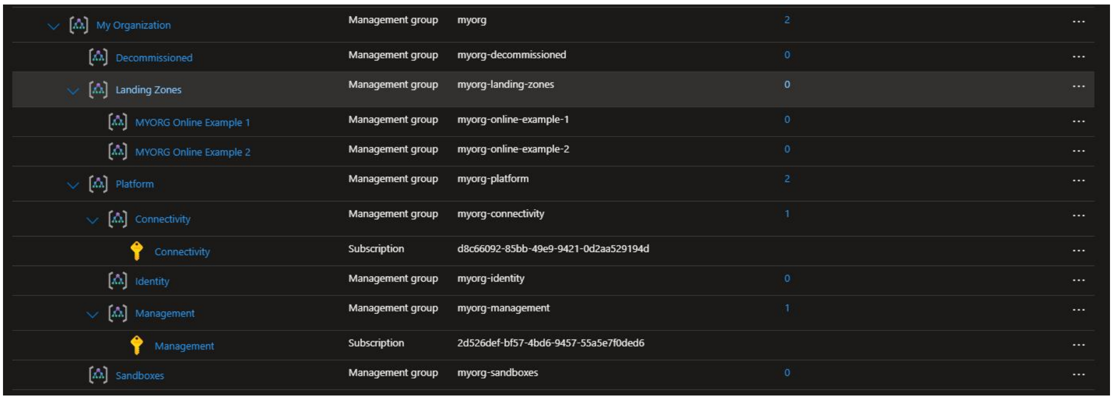


```
sarang [ ~/remote/management ]$ cat variables.tf
# Use variables to customize the deployment

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

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

variable "subscription_id_management" {
  type        = string
  description = "Subscription ID to use for \"management\" resources."
  default = "2d526def-bf57-4bd6-9457-55a5e7f0ded6"
}

variable "email_security_contact" {
  type        = string
  description = "Set a custom value for the security contact email address."
  default     = "pardeshisarang717@gmail.com"
}

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 "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"
  }
}
```

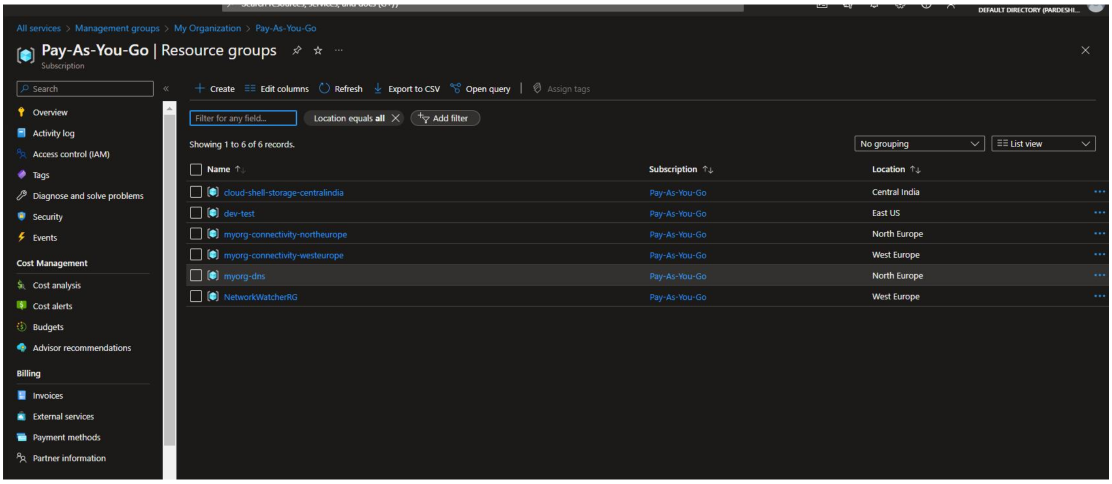
## Observations/Descriptions:

1. Firstly, we applied the code for connectivity and management as they consisted of resources and separate subscriptions.
2. Then we applied the code for core which has the archetype.json file for the landing zone and did not require a subscription.
3. The connectivity resources are properly being created and allocated to the connectivity management group with the connectivity subscription and same for management.
4. On successful completion of deployment we get resources, solutions and landing zones with policies with different state files.

## Below are the results of the deployment



| | | | |
|---|---|---|---|
| My Organization | Management group | myorg | 2 |
| 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 | 2 |
| Connectivity | Management group | myorg-connectivity | 1 |
| Connectivity | Subscription | d8c66092-85bb-49e9-9421-0d2aa529194d | |
| Identity | Management group | myorg-identity | 0 |
| Management | Management group | myorg-management | 1 |
| Management | Subscription | 2d526def-bf57-4bd6-9457-55a5e7f0ded6 | |
| Sandboxes | Management group | myorg-sandboxes | 0 |



Pay-As-You-Go | Resource groups

Subscription

Showing 1 to 6 of 6 records.

| Name | Subscription | Location |
|---|---|---|
| cloud-shell-storage-centralindia | Pay-As-You-Go | Central India |
| dev-test | Pay-As-You-Go | East US |
| 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 |
| NetworkWatcherRG | Pay-As-You-Go | West Europe |

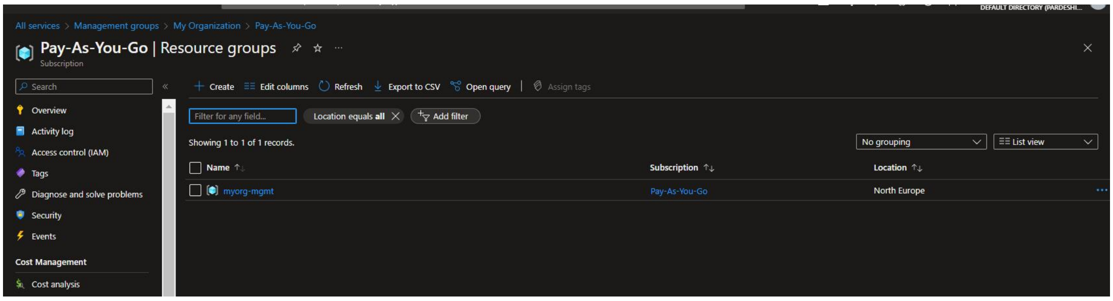
## List of Resources in connectivity

| NAME | TYPE | RESOURCE GROUP | LOCATION | SUBSCRIPTION |
|---|---|---|---|---|
| myorg-hub-northeurope | Virtual network | myorg-connectivity-northeurope | North Europe | Pay-As-You-Go |
| myorg-hub-westeurope | Virtual | myorg-connectivity- | West | Pay-As- |

| | | | | |
|---|---|---|---|---|
| | network | westeurope | Europe | You-Go |
| NetworkWatcher_northeurope | Network Watcher | networkwatcherrg | North Europe | Pay-As-You-Go |
| NetworkWatcher_westeurope | Network Watcher | networkwatcherrg | West Europe | Pay-As-You-Go |
| privatelink.adf.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.afs.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.agentsvc.azure-automation.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.analysis.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.api.azureml.ms | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azconfig.io | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azure-api.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azure-automation.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azure-devices-provisioning.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azure-devices.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azurecr.io | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azurehdinsight.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azurehealthcareapis.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azurestaticapps.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azuresynapse.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.azurewebsites.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.batch.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.blob.core.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.cassandra.cosmos.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.cognitiveservices.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.database.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.datafactory.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.dev.azuresynapse.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |

| | | | | |
|---|---|---|---|---|
| privatelink.developer.azure-api.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.dfs.core.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.dicom.azurehealthcareapis.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.digitaltwins.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.directline.botframework.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.documents.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.eventgrid.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.file.core.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.gremlin.cosmos.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.guestconfiguration.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.his.arc.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.kubernetesconfiguration.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.managedhsm.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.mariadb.database.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.media.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.mongo.cosmos.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.monitor.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.mysql.database.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.ne.backup.windowsazure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.northeurope.azmk8s.io | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.northeurope.kusto.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.notebooks.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.ods.opinsights.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.oms.opinsights.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.pbidedicated.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.postgres.database. | Private | myorg-dns | Global | Pay-As- |

| | | | | |
|---|---|---|---|---|
| azure.com | DNS zone | | | You-Go |
| privatelink.prod.migration.windowsazure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.purview.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.purviewstudio.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.queue.core.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.redis.cache.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.redisenterprise.cache.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.search.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.service.signalr.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.servicebus.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.siterecovery.windowsazure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.sql.azuresynapse.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.table.core.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.table.cosmos.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.tip1.powerquery.microsoft.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.token.botframework.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.vaultcore.azure.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.web.core.windows.net | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |
| privatelink.webpubsub.azure.com | Private DNS zone | myorg-dns | Global | Pay-As-You-Go |

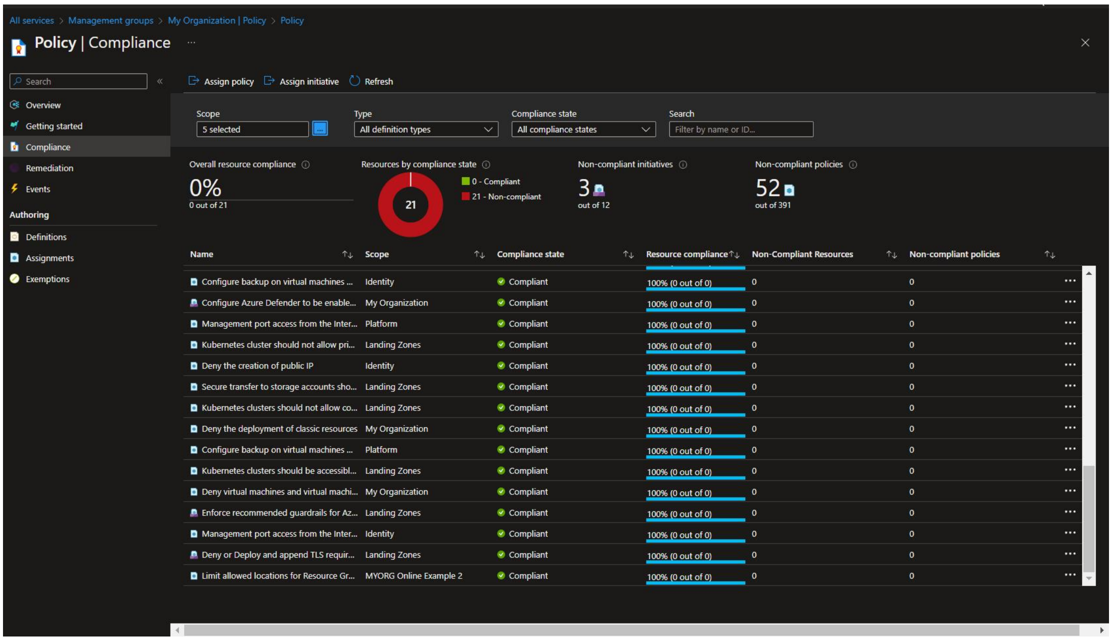


Showing 1 to 1 of 1 records.

| Name | Subscription | Location |
|---|---|---|
| myorg-mgmt | Pay-As-You-Go | North Europe |

## List of Resources in Management

| NAME | TYPE | RESOURCE GROUP | LOCATION | SUBSCRIPTION |
|---|---|---|---|---|
| AgentHealthAssessment(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| AntiMalware(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| ChangeTracking(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| ContainerInsights(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| myorg-automation | Automation Account | myorg-mgmt | North Europe | Pay-As-You-Go |
| myorg-la | Log Analytics workspace | myorg-mgmt | North Europe | Pay-As-You-Go |
| Security(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| SecurityInsights(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| SQLAdvancedThreatProtection(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| SQLAssessment(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| SQLVulnerabilityAssessment(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| Updates(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |
| VMInsights(myorg-la) | Solution | myorg-mgmt | North Europe | Pay-As-You-Go |



Scope: 5 selected
Type: All definition types
Compliance state: All compliance states
Search: Filter by name or ID...

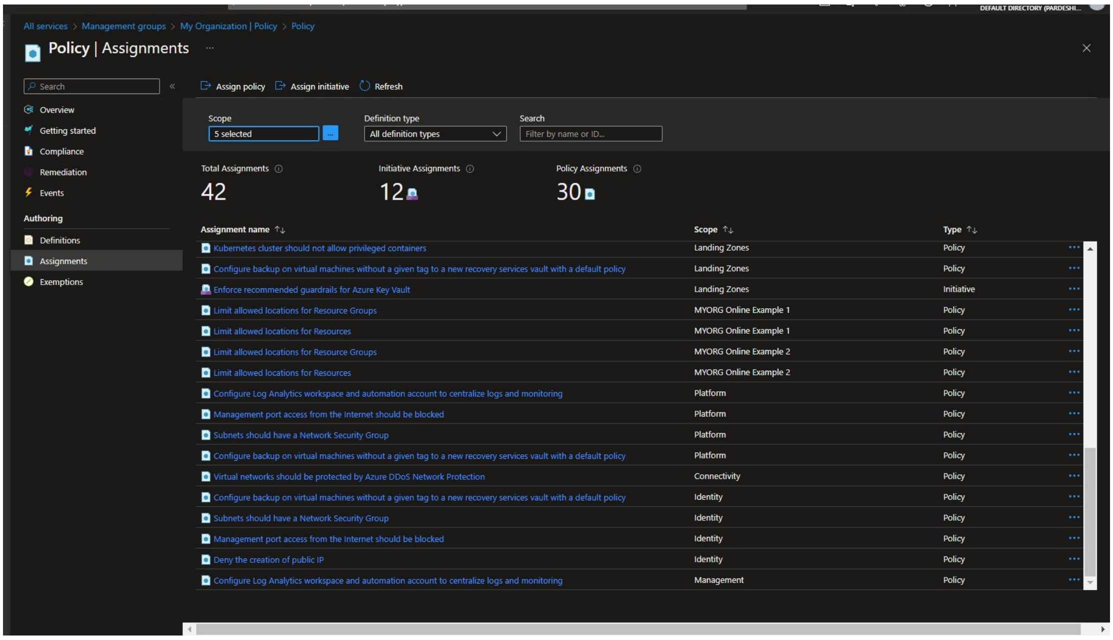
Overall resource compliance: 0%
0 out of 21

Resources by compliance state: 21
🟩 0 - Compliant
🟥 21 - Non-compliant

Non-compliant initiatives: 3
out of 12

Non-compliant policies: 52
out of 391

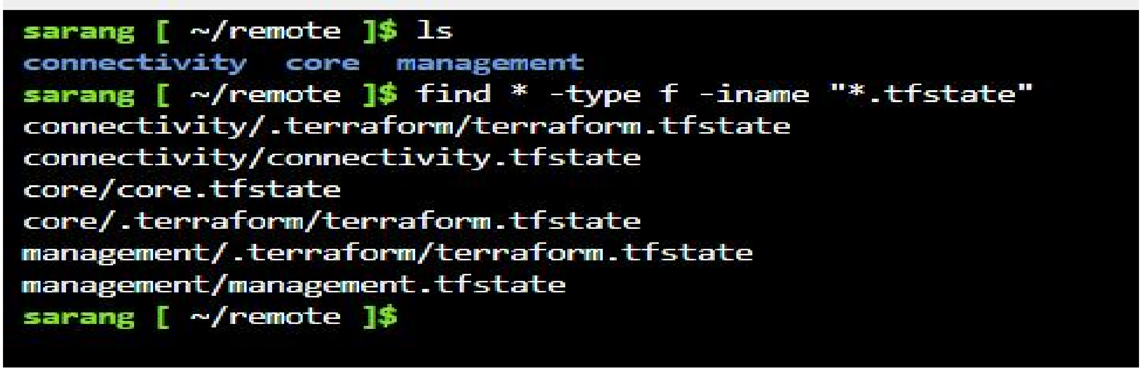
| Name | Scope | Compliance state | Resource compliance | Non-Compliant Resources | Non-compliant policies |
|---|---|---|---|---|---|
| Configure backup on virtual machines ... | Identity | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Configure Azure Defender to be enable... | My Organization | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Management port access from the Inter... | Platform | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Kubernetes cluster should not allow pri... | Landing Zones | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Deny the creation of public IP | Identity | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Secure transfer to storage accounts sho... | Landing Zones | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Kubernetes clusters should not allow co... | Landing Zones | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Deny the deployment of classic resources | My Organization | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Configure backup on virtual machines ... | Platform | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Kubernetes clusters should be accessibl... | Landing Zones | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Deny virtual machines and virtual machi... | My Organization | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Enforce recommended guardrails for Az... | Landing Zones | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Management port access from the Inter... | Identity | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Deny or Deploy and append TLS requir... | Landing Zones | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |
| Limit allowed locations for Resource Gr... | MYORG Online Example 2 | ✓ Compliant | 100% (0 out of 0) | 0 | 0 |



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 |
|---|---|---|
| Kubernetes cluster should not allow privileged containers | Landing Zones | Policy |
| Configure backup on virtual machines without a given tag to a new recovery services vault with a default policy | Landing Zones | Policy |
| Enforce recommended guardrails for Azure Key Vault | Landing Zones | Initiative |
| Limit allowed locations for Resource Groups | MYORG Online Example 1 | Policy |
| Limit allowed locations for Resources | MYORG Online Example 1 | Policy |
| Limit allowed locations for Resource Groups | MYORG Online Example 2 | Policy |
| Limit allowed locations for Resources | MYORG Online Example 2 | Policy |
| Configure Log Analytics workspace and automation account to centralize logs and monitoring | Platform | Policy |
| Management port access from the Internet should be blocked | Platform | Policy |
| Subnets should have a Network Security Group | Platform | Policy |
| Configure backup on virtual machines without a given tag to a new recovery services vault with a default policy | Platform | Policy |
| Virtual networks should be protected by Azure DDoS Network Protection | Connectivity | Policy |
| Configure backup on virtual machines without a given tag to a new recovery services vault with a default policy | Identity | Policy |
| Subnets should have a Network Security Group | Identity | Policy |
| Management port access from the Internet should be blocked | Identity | Policy |
| Deny the creation of public IP | Identity | Policy |
| Configure Log Analytics workspace and automation account to centralize logs and monitoring | Management | Policy |



```
sarang [ ~/remote ]$ ls
connectivity  core  management
sarang [ ~/remote ]$ find * -type f -iname "*.tfstate"
connectivity/.terraform/terraform.tfstate
connectivity/connectivity.tfstate
core/core.tfstate
core/.terraform/terraform.tfstate
management/.terraform/terraform.tfstate
management/management.tfstate
sarang [ ~/remote ]$
```