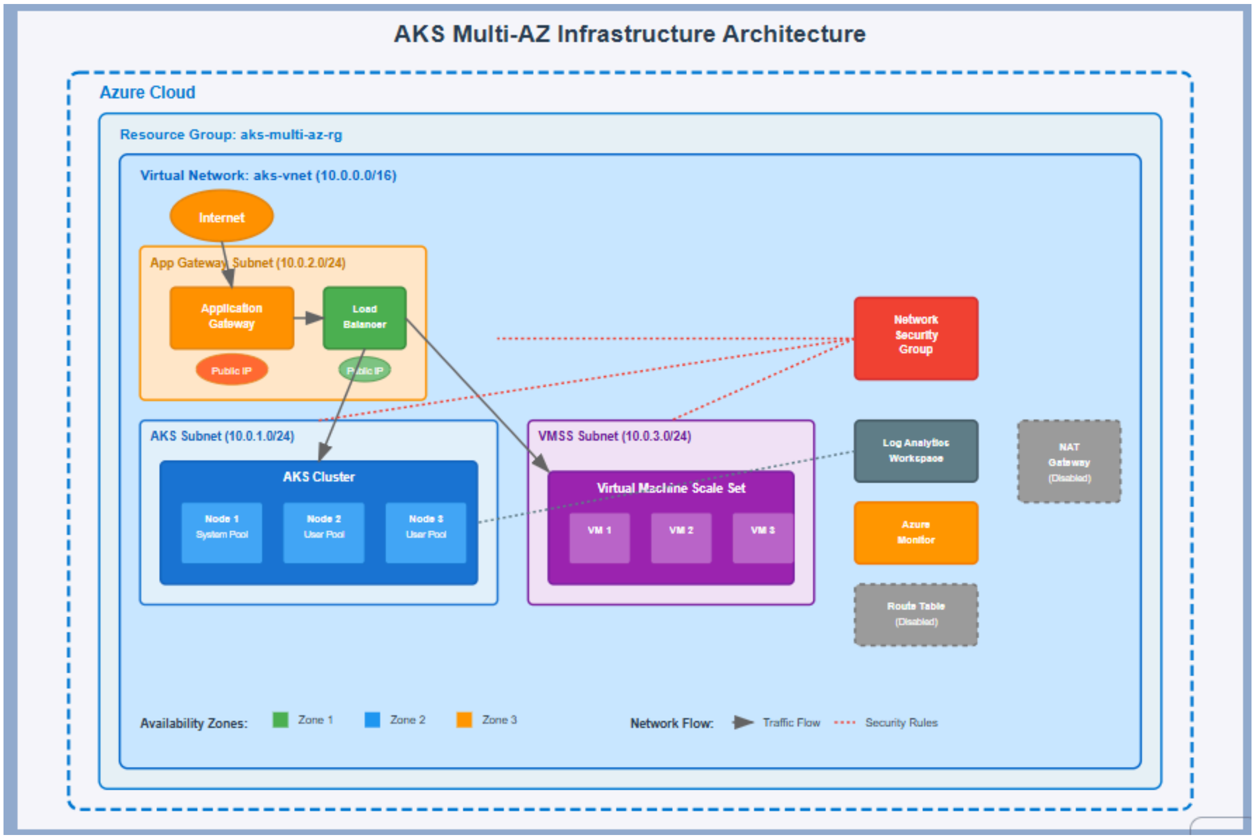
cprime

Vadakathi Muhammed Suhaib
Technical Apprentice
Emp ID: X48GRSTML
[*muhammed.suhaib@cprime.com*](mailto:muhammed.suhaib@cprime.com)

Setup a 3-Node Multi-Zone AKS Cluster Using Terraform



Phase 2: Infrastructure Setup Using Terraform

Table of Contents

Prerequisites

System Requirements

- **Operating System**: Debian WSL2 on Windows
- **Azure Subscription**: Active Azure subscription with appropriate permissions
- **Tools Required**: curl, unzip, gnupg, lsb-release

Azure Permissions Required

- Contributor role on the target Azure subscription
- Ability to create service principals
- Access to create resources in the target region

Phase 1: Environment Setup

Step 1: Update System and Install Dependencies

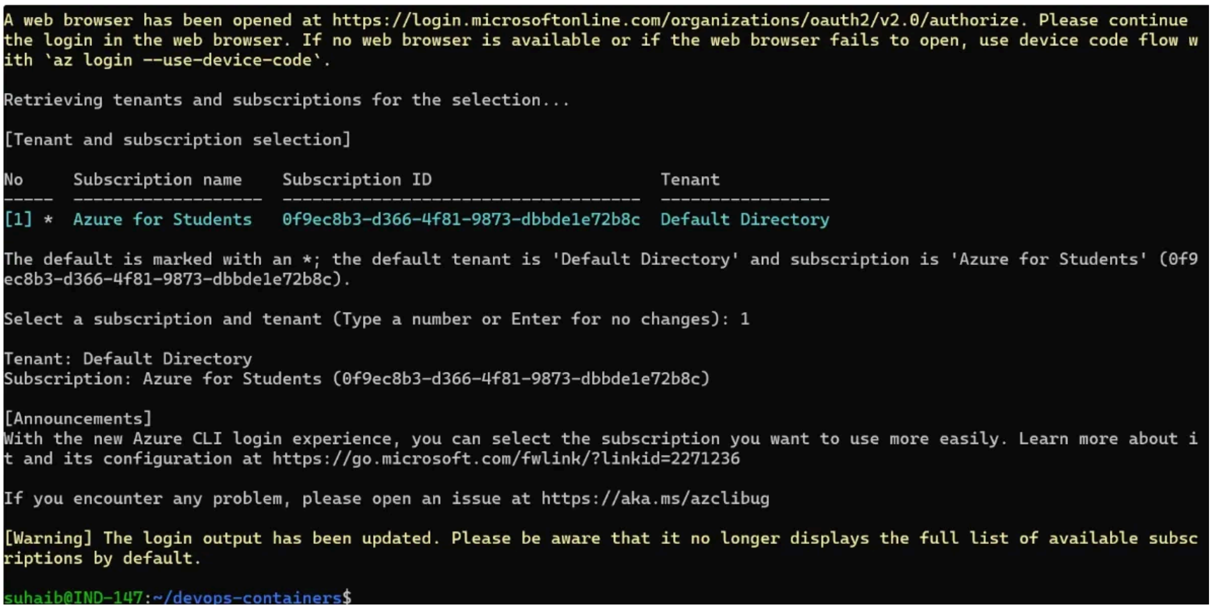
```
# Update package list and upgrade system
sudo apt update && sudo apt upgrade -y
```

```
# Install required packages
sudo apt install -y curl unzip gnupg lsb-release
```

Step 2: Install Azure CLI

```
# Add Azure CLI repository and install
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

# Verify installation
az --version

# Login to Azure
az login
```

Expected Output: Browser-based authentication will open. Complete the login process.



```
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue
the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow w
ith `az login --use-device-code`.

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No     Subscription name    Subscription ID                       Tenant
-----  -------------------  ------------------------------------  -----------------
[1] *  Azure for Students   0f9ec8b3-d366-4f81-9873-dbbde1e72b8c  Default Directory

The default is marked with an *; the default tenant is 'Default Directory' and subscription is 'Azure for Students' (0f9
ec8b3-d366-4f81-9873-dbbde1e72b8c).

Select a subscription and tenant (Type a number or Enter for no changes): 1

Tenant: Default Directory
Subscription: Azure for Students (0f9ec8b3-d366-4f81-9873-dbbde1e72b8c)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about i
t and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subsc
riptions by default.

suhaib@IND-147:~/devops-containers$
```

Step 3: Install Terraform

```
# Download Terraform (using version 1.12.1 as per documentation)
wget https://releases.hashicorp.com/terraform/1.12.1/terraform_1.12.1_linux_amd64.zip
```

```
# Unzip and install
unzip terraform_1.12.1_linux_amd64.zip
sudo mv terraform /usr/local/bin/

# Verify installation
terraform version
```

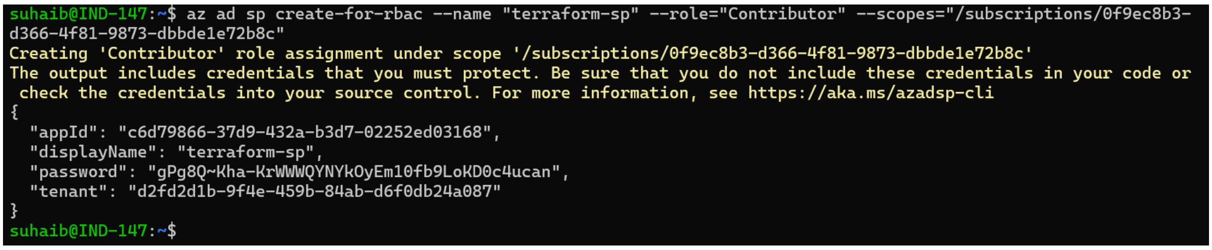
Expected Output: `Terraform v1.12.1`

Step 4: Create Azure Service Principal

```
# Get your subscription ID
az account show --query id -o tsv

# Create service principal (replace SUBSCRIPTION_ID with your actual ID)
az ad sp create-for-rbac \
  --name "terraform-sp" \
  --role="Contributor" \
  --scopes="/subscriptions/SUBSCRIPTION_ID"
```

Sample Output:

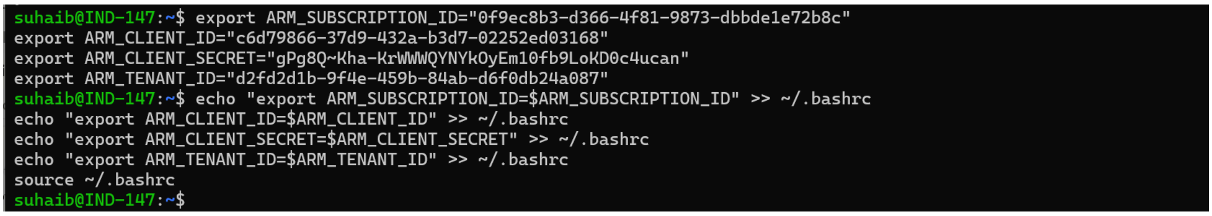
```
{
  "appId": "c6d7986637d9432a-b3d702252ed03168",
  "displayName": "terraform-sp",
  "password": "gPg8QKha-KrWWWQYNYkOyEm10fb9LoKD0c4ucan",
  "tenant": "d2fd2d1b-9f4e-459b-84ab-d6f0db24a087"
}
```



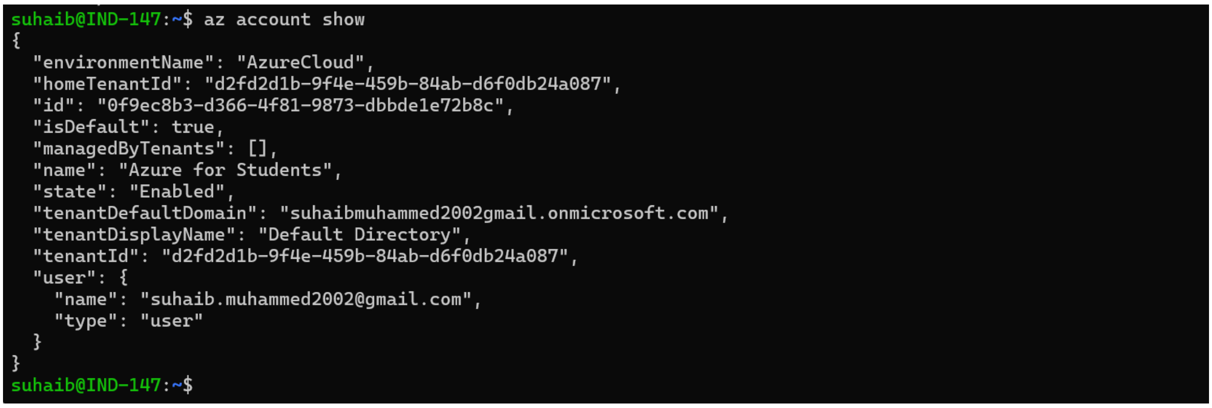
```
suhaib@IND-147:~$ az ad sp create-for-rbac --name "terraform-sp" --role="Contributor" --scopes="/subscriptions/0f9ec8b3-
d366-4f81-9873-dbbde1e72b8c"
Creating 'Contributor' role assignment under scope '/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c'
The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or
 check the credentials into your source control. For more information, see https://aka.ms/azadsp-cli
{
  "appId": "c6d79866-37d9-432a-b3d7-02252ed03168",
  "displayName": "terraform-sp",
  "password": "gPg8Q~Kha-KrWWWQYNYkOyEm10fb9LoKD0c4ucan",
  "tenant": "d2fd2d1b-9f4e-459b-84ab-d6f0db24a087"
}
suhaib@IND-147:~$
```

Step 5: Set Environment Variables

```
# Export credentials for current session
export ARM_SUBSCRIPTION_ID="your-subscription-id"
export ARM_CLIENT_ID="your-app-id"
export ARM_CLIENT_SECRET="your-password"
export ARM_TENANT_ID="your-tenant-id"

# Make variables persistent
echo "export ARM_SUBSCRIPTION_ID=\"$ARM_SUBSCRIPTION_ID\"" >> ~/.bashrc
echo "export ARM_CLIENT_ID=\"$ARM_CLIENT_ID\"" >> ~/.bashrc
echo "export ARM_CLIENT_SECRET=\"$ARM_CLIENT_SECRET\"" >> ~/.bashrc
echo "export ARM_TENANT_ID=\"$ARM_TENANT_ID\"" >> ~/.bashrc

# Reload bashrc
source ~/.bashrc

# Verify authentication
az account show
```



```
suhaib@IND-147:~$ export ARM_SUBSCRIPTION_ID="0f9ec8b3-d366-4f81-9873-dbbde1e72b8c"
export ARM_CLIENT_ID="c6d79866-37d9-432a-b3d7-02252ed03168"
export ARM_CLIENT_SECRET="gPg8Q~Kha-KrWWWQYNYkOyEm10fb9LoKD0c4ucan"
export ARM_TENANT_ID="d2fd2d1b-9f4e-459b-84ab-d6f0db24a087"
suhaib@IND-147:~$ echo "export ARM_SUBSCRIPTION_ID=$ARM_SUBSCRIPTION_ID" >> ~/.bashrc
echo "export ARM_CLIENT_ID=$ARM_CLIENT_ID" >> ~/.bashrc
echo "export ARM_CLIENT_SECRET=$ARM_CLIENT_SECRET" >> ~/.bashrc
echo "export ARM_TENANT_ID=$ARM_TENANT_ID" >> ~/.bashrc
source ~/.bashrc
suhaib@IND-147:~$
```



```
suhaib@IND-147:~$ az account show
{
  "environmentName": "AzureCloud",
  "homeTenantId": "d2fd2d1b-9f4e-459b-84ab-d6f0db24a087",
  "id": "0f9ec8b3-d366-4f81-9873-dbbde1e72b8c",
  "isDefault": true,
  "managedByTenants": [],
  "name": "Azure for Students",
  "state": "Enabled",
  "tenantDefaultDomain": "suhaibmuhammed2002gmail.onmicrosoft.com",
  "tenantDisplayName": "Default Directory",
  "tenantId": "d2fd2d1b-9f4e-459b-84ab-d6f0db24a087",
  "user": {
    "name": "suhaib.muhammed2002@gmail.com",
    "type": "user"
  }
}
suhaib@IND-147:~$
```

Step 6: Create Project Directory

```
mkdir ~/terraform-aks-multi-az
cd ~/terraform-aks-multi-az
```

Phase 2: Infrastructure Setup Using Terraform

Project Structure Overview

```
terraform-aks-multi-az/
├── modules/
│   ├── vnet/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── subnets/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── nsg/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── route_table/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── nat_gateway/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── app_gateway/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── load_balancer/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── vmss/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   └── aks/
│       ├── main.tf
│       ├── variables.tf
│       └── outputs.tf
├── main.tf
├── variables.tf
├── outputs.tf
├── terraform.tfvars
└── README.md
```

Step 1: Create Directory Structure

```
# Create module directories
mkdir -p terraform-aks-multi-az/modules/{vnet,subnets,nsg,route_table,nat_gateway,app_gateway,load_balancer,vmss,aks}

# Create module files
for module in vnet subnets nsg route_table nat_gateway app_gateway load_balancer vmss aks; do
  touch terraform-aks-multi-az/modules/$module/{main.tf,variables.tf,outputs.tf}
done

# Create root module files
touch terraform-aks-multi-az/{main.tf,variables.tf,outputs.tf,terraform.tfvars,README.md}
```

Step 2: Configure Terraform Modules

Module 1: VNET

main.tf:

```
resource "azurerm_virtual_network" "vnet" {
  name                = var.vnet_name
  address_space       = var.address_space
  location            = var.location
  resource_group_name = var.resource_group_name
  tags                = var.tags
}
```

variables.tf:

```
variable "vnet_name" {
  description = "Name of the virtual network"
  type        = string
}

variable "address_space" {
  description = "Address space for the VNET"
  type        = list(string)
}

variable "location" {
  description = "Azure region"
  type        = string
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

outputs.tf:

```
output "vnet_id" {
  description = "ID of the VNET"
  value       = azurerm_virtual_network.vnet.id
}

output "vnet_name" {
  description = "Name of the VNET"
  value       = azurerm_virtual_network.vnet.name
}
```

Module 2: Subnets

main.tf:

```
# modules/subnets/main.tf
resource "azurerm_subnet" "subnets" {
  for_each             = var.subnets
  name                 = each.key
  resource_group_name  = var.resource_group_name
  virtual_network_name = var.vnet_name
  address_prefixes     = [each.value.address_prefix]

  # Service endpoints for AKS and Application Gateway
  service_endpoints = each.key == "aks" ? ["Microsoft.Storage", "Microsoft.KeyVault"] : (
    each.key == "appgateway" ? ["Microsoft.Web"] : []
  )

  # Remove delegation block - Application Gateway doesn't require subnet delegation
  # Application Gateway can be deployed to any subnet without special delegation
}
```

variables.tf:

```
variable "subnets" {
  description = "Map of subnet names to address prefixes"
  type        = map(object({
    address_prefix = string
  }))
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "vnet_name" {
  description = "Name of the virtual network"
  type        = string
}
```

outputs.tf:

```
output "subnet_ids" {
  description = "Map of subnet names to their IDs"
  value       = { for k, v in azurerm_subnet.subnets : k => v.id }
}
```

Module 3: Network Security Groups (NSGs)

main.tf:

```
# Fixed modules/nsg/main.tf - Application Gateway v2 Compatible
resource "azurerm_network_security_group" "nsg" {
  name                = var.nsg_name
  location            = var.location
  resource_group_name = var.resource_group_name
  tags                = var.tags

  # Allow HTTP traffic
  security_rule {
    name                       = "allow-http"
    priority                   = 1000
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "80"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }

  # Allow HTTPS traffic
  security_rule {
    name                       = "allow-https"
    priority                   = 1010
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "443"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }

  # Allow SSH traffic
  security_rule {
    name                       = "allow-ssh"
    priority                   = 1020
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "22"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }

  # Allow AKS API server traffic
```

```
security_rule {
  name                       = "allow-aks-api"
  priority                   = 1030
  direction                  = "Inbound"
  access                     = "Allow"
  protocol                   = "Tcp"
  source_port_range          = "*"
  destination_port_range     = "443"
  source_address_prefix      = "AzureCloud"
  destination_address_prefix = "*"
}

# Allow internal subnet communication
security_rule {
  name                       = "allow-internal"
  priority                   = 1040
  direction                  = "Inbound"
  access                     = "Allow"
  protocol                   = "*"
  source_port_range          = "*"
  destination_port_range     = "*"
  source_address_prefix      = "10.0.0.0/16"
  destination_address_prefix = "10.0.0.0/16"
}

# CRITICAL: Allow Application Gateway v2 management ports
security_rule {
  name                       = "allow-appgw-management"
  priority                   = 1050
  direction                  = "Inbound"
  access                     = "Allow"
  protocol                   = "Tcp"
  source_port_range          = "*"
  destination_port_range     = "65200-65535"
  source_address_prefix      = "GatewayManager"
  destination_address_prefix = "*"
}
```

```
  # Allow Azure Load Balancer health probes
  security_rule {
    name                       = "allow-lb-probe"
    priority                   = 1060
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "*"
    source_port_range          = "*"
    destination_port_range     = "*"
    source_address_prefix      = "AzureLoadBalancer"
    destination_address_prefix = "*"
  }
}
```

variables.tf:

```
variable "nsg_name" {
  description = "Name of the NSG"
  type        = string
}

variable "location" {
  description = "Azure region"
  type        = string
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

outputs.tf:

```
output "nsg_id" {
  description = "ID of the NSG"
  value       = azurerm_network_security_group.nsg.id
}
```

Module 4: Route Tables

main.tf:

```
# modules/route_table/main.tf - Fixed with proper subnet association
resource "azurerm_route_table" "route_table" {
  name                = var.route_table_name
  location            = var.location
  resource_group_name = var.resource_group_name
  tags                = var.tags

  # Use the new property name instead of deprecated one
  bgp_route_propagation_enabled = true
}

resource "azurerm_route" "routes" {
  for_each               = var.routes
  name                   = each.key
  resource_group_name    = var.resource_group_name
  route_table_name       = azurerm_route_table.route_table.name
  address_prefix         = each.value.address_prefix
  next_hop_type          = each.value.next_hop_type
  next_hop_in_ip_address = each.value.next_hop_in_ip_address != null ? each.
}

# Associate route table with AKS subnet - This is critical for AKS with userDef
resource "azurerm_subnet_route_table_association" "aks" {
  count          = var.associate_with_subnets ? 1 : 0
  subnet_id      = var.aks_subnet_id
  route_table_id = azurerm_route_table.route_table.id
  depends_on = [
    azurerm_route_table.route_table,
    azurerm_route.routes
  ]
}
```

variables.tf:

```
variable "route_table_name" {
  description = "Name of the route table"
  type        = string
}

variable "location" {
  description = "Azure region"
  type        = string
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "routes" {
  description = "Map of routes"
  type        = map(object({
    address_prefix         = string
    next_hop_type          = string
    next_hop_in_ip_address = optional(string)
  }))
  default = {}
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

```
variable "associate_with_subnets" {
  description = "Whether to associate route table with subnets"
  type        = bool
  default     = false
}

variable "aks_subnet_id" {
  description = "AKS subnet ID for route table association"
  type        = string
  default     = ""
}
```

outputs.tf:

```
output "route_table_id" {
  description = "ID of the route table"
  value       = azurerm_route_table.route_table.id
}
```

Module 5: NAT Gateway

main.tf:

```
# modules/nat_gateway/main.tf - Fixed version
resource "azurerm_public_ip" "nat_ip" {
  name                = "${var.nat_gateway_name}-ip"
  location            = var.location
  resource_group_name = var.resource_group_name
  allocation_method   = "Static"
  sku                 = "Standard"
  zones               = ["3"]  # Specify zone for consistency
  tags                = var.tags
}

resource "azurerm_nat_gateway" "nat_gateway" {
  name                    = var.nat_gateway_name
  location                = var.location
  resource_group_name     = var.resource_group_name
  sku_name                = "Standard"
  idle_timeout_in_minutes = 10
  zones                   = ["3"]  # Specify zone for consistency
  tags                    = var.tags

  depends_on = [
    azurerm_public_ip.nat_ip
  ]
}

resource "azurerm_nat_gateway_public_ip_association" "nat_ip_assoc" {
  nat_gateway_id       = azurerm_nat_gateway.nat_gateway.id
  public_ip_address_id = azurerm_public_ip.nat_ip.id

  depends_on = [
    azurerm_nat_gateway.nat_gateway,
    azurerm_public_ip.nat_ip
  ]
}
```

variables.tf:

```
variable "nat_gateway_name" {
  description = "Name of the NAT Gateway"
  type        = string
}

variable "location" {
  description = "Azure region"
  type        = string
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}
```

```
variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

outputs.tf:

```
output "nat_gateway_id" {
  description = "ID of the NAT Gateway"
  value       = azurerm_nat_gateway.nat_gateway.id
}
```

Module 6: Application Gateway

main.tf:

```
resource "azurerm_public_ip" "app_gw_ip" {
  name                = "${var.app_gateway_name}-ip"
  location            = var.location
  resource_group_name = var.resource_group_name
  allocation_method   = "Static"
  sku                 = "Standard"
  tags                = var.tags
}

resource "azurerm_application_gateway" "app_gateway" {
  name                = var.app_gateway_name
  resource_group_name = var.resource_group_name
  location            = var.location
  tags                = var.tags

  sku {
    name     = "Standard_v2"
    tier     = "Standard_v2"
    capacity = 2
  }

  gateway_ip_configuration {
    name      = "app-gateway-ip-config"
    subnet_id = var.subnet_id
  }

  frontend_port {
    name = "frontend-port"
    port = 80
  }

  frontend_ip_configuration {
    name                 = "frontend-ip-config"
    public_ip_address_id = azurerm_public_ip.app_gw_ip.id
  }

  backend_address_pool {
    name = "backend-pool"
  }

  backend_http_settings {
    name                  = "backend-http-settings"
    cookie_based_affinity = "Disabled"
    port                  = 80
    protocol              = "Http"
    request_timeout       = 20
  }

  http_listener {
    name                           = "http-listener"
    frontend_ip_configuration_name = "frontend-ip-config"
    frontend_port_name             = "frontend-port"
    protocol                       = "Http"
  }

  request_routing_rule {
    name                       = "routing-rule"
    rule_type                  = "Basic"
    priority                   = 1000
    http_listener_name         = "http-listener"
    backend_address_pool_name  = "backend-pool"
    backend_http_settings_name = "backend-http-settings"
  }
}
```

variables.tf:

```
variable "app_gateway_name" {
  description = "Name of the Application Gateway"
  type        = string
}

variable "location" {
  description = "Azure region"
  type        = string
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "subnet_id" {
  description = "ID of the subnet for the Application Gateway"
  type        = string
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

outputs.tf:

```
output "app_gateway_id" {
  description = "ID of the Application Gateway"
  value       = azurerm_application_gateway.app_gateway.id
}
```

Module 7: Load Balancer

main.tf:

```
# modules/load_balancer/main.tf - Fixed version
resource "azurerm_public_ip" "lb_ip" {
  name                = "${var.lb_name}-ip"
  location            = var.location
  resource_group_name = var.resource_group_name
  allocation_method   = "Static"
  sku                 = "Standard"
  zones               = ["3"]  # Specify zone for consistency
  tags                = var.tags
}

resource "azurerm_lb" "load_balancer" {
  name                = var.lb_name
  location            = var.location
  resource_group_name = var.resource_group_name
  sku                 = "Standard"
  sku_tier            = "Regional"
  tags                = var.tags

  frontend_ip_configuration {
    name                 = "frontend-ip-config"
    public_ip_address_id = azurerm_public_ip.lb_ip.id
  }

  depends_on = [
    azurerm_public_ip.lb_ip
  ]
}
```

```
resource "azurerm_lb_backend_address_pool" "backend_pool" {
  loadbalancer_id = azurerm_lb.load_balancer.id
  name            = "backend-pool"
}

resource "azurerm_lb_probe" "probe" {
  loadbalancer_id = azurerm_lb.load_balancer.id
  name            = "http-probe"
  protocol        = "Http"
  port            = 80
  request_path    = "/"
}

resource "azurerm_lb_rule" "rule" {
  loadbalancer_id                = azurerm_lb.load_balancer.id
  name                           = "http-rule"
  protocol                       = "Tcp"
  frontend_port                  = 80
  backend_port                   = 80
  frontend_ip_configuration_name = "frontend-ip-config"
  backend_address_pool_ids       = [azurerm_lb_backend_address_pool.backen
  probe_id                       = azurerm_lb_probe.probe.id
  disable_outbound_snat          = true
}
```

variables.tf:

```
variable "lb_name" {
  description = "Name of the Load Balancer"
  type        = string
}

variable "location" {
  description = "Azure region"
  type        = string
}
```

```
variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

outputs.tf:

```
output "lb_id" {
  description = "ID of the Load Balancer"
  value       = azurerm_lb.load_balancer.id
}

output "backend_pool_id" {
  description = "ID of the backend address pool"
  value       = azurerm_lb_backend_address_pool.backend_pool.id
}
```

Module 8: Virtual Machine Scale Set (VMSS)

main.tf:

```
# Fixed modules/vmss/main.tf - Remove zones to avoid allocation issues
resource "azurerm_linux_virtual_machine_scale_set" "vmss" {
  name                = var.vmss_name
  resource_group_name = var.resource_group_name
  location            = var.location
  sku                 = var.vm_size
  instances           = var.instance_count
  admin_username      = var.admin_username
  tags                = var.tags
  # Disable password authentication
  disable_password_authentication = true

  # Remove zones to avoid allocation issues - let Azure choose best placement
  # zones = ["3"]

  source_image_reference {
    publisher = "Canonical"
    offer     = "0001-com-ubuntu-server-focal"
    sku       = "20_04-lts-gen2"
    version   = "latest"
  }

  os_disk {
    storage_account_type = "Standard_LRS"
    caching              = "ReadWrite"
  }

  network_interface {
    name    = "${var.vmss_name}-nic"
    primary = true

    ip_configuration {
      name                                   = "internal"
      subnet_id                              = var.subnet_id
      primary                                = true
      load_balancer_backend_address_pool_ids = var.backend_pool_id != null ? [var.backend_pool_id] : []
    }
  }

  admin_ssh_key {
    username   = var.admin_username
    public_key = file(var.ssh_public_key_path)
  }

  # Custom script extension for basic setup
  extension {
    name                 = "HealthExtension"
    publisher            = "Microsoft.ManagedServices"
    type                 = "ApplicationHealthLinux"
    type_handler_version = "1.0"
    settings = jsonencode({
      protocol    = "http"
      port        = 80
      requestPath = "/"
    })
  }
}
```

variables.tf:

```
variable "vmss_name" {
  description = "Name of the VMSS"
  type        = string
}

variable "location" {
  description = "Azure region"
  type        = string
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "vm_size" {
  description = "VM size for the scale set"
  type        = string
}

variable "instance_count" {
  description = "Number of VM instances"
  type        = number
}

variable "admin_username" {
  description = "Admin username for VMs"
  type        = string
}

variable "ssh_public_key_path" {
  description = "Path to the SSH public key"
  type        = string
}

variable "subnet_id" {
  description = "ID of the subnet for the VMSS"
  type        = string
}

variable "backend_pool_id" {
  description = "ID of the load balancer backend pool"
  type        = string
  default     = null
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

outputs.tf:

```
output "vmss_id" {
  description = "ID of the VMSS"
  value       = azurerm_linux_virtual_machine_scale_set.vmss.id
}
```

Module 9: AKS

main.tf:

```
# Fixed modules/aks/main.tf - Remove userDefinedRouting for student subscr
# Log Analytics Workspace for AKS monitoring - Create this first
resource "azurerm_log_analytics_workspace" "aks" {
  name                = "${var.aks_name}-logs"
  location            = var.location
  resource_group_name = var.resource_group_name
  sku                 = "PerGB2018"
  retention_in_days   = 30
  tags                = var.tags
}

resource "azurerm_kubernetes_cluster" "aks" {
  name                = var.aks_name
  location            = var.location
  resource_group_name = var.resource_group_name
  dns_prefix          = "${var.aks_name}-dns"
  kubernetes_version  = var.kubernetes_version
  tags                = var.tags

  # Enable RBAC
  role_based_access_control_enabled = true

  # Enable local accounts for student subscription
  local_account_disabled = false

  default_node_pool {
    name                = "system"
    node_count          = var.node_count
    vm_size             = var.vm_size
    enable_auto_scaling = true
    min_count           = var.node_count
    max_count           = var.node_count + 1  # Reduced for student subscription
    vnet_subnet_id      = var.subnet_id
    type                = "VirtualMachineScaleSets"
    os_disk_size_gb     = 30
    os_disk_type        = "Managed"
    # Node labels for system pool
    node_labels = {
      "nodepool-type" = "system"
      "environment"   = "development"
    }
  }

  identity {
    type = "SystemAssigned"
  }

  # SIMPLIFIED network profile - Remove userDefinedRouting for student subs
  network_profile {
    network_plugin      = "azure"
    network_policy      = "azure"
    load_balancer_sku   = "standard"
    # CHANGED: Use loadBalancer instead of userDefinedRouting
    outbound_type       = "loadBalancer"
    service_cidr        = "172.16.0.0/16"
    dns_service_ip      = "172.16.0.10"
  }

  # Enable monitoring with explicit dependency
  oms_agent {
    log_analytics_workspace_id = azurerm_log_analytics_workspace.aks.id
  }

  # Add explicit dependency
  depends_on = [
    azurerm_log_analytics_workspace.aks
  ]
}
```

variables.tf:

```
variable "aks_name" {
  description = "Name of the AKS cluster"
  type        = string
}

variable "location" {
  description = "Azure region"
  type        = string
}

variable "resource_group_name" {
  description = "Resource group name"
  type        = string
}

variable "kubernetes_version" {
  description = "Kubernetes version"
  type        = string
}

variable "node_count" {
  description = "Number of nodes in the default node pool"
  type        = number
}

variable "vm_size" {
  description = "VM size for the node pool"
  type        = string
}

variable "subnet_id" {
  description = "ID of the subnet for the AKS cluster"
  type        = string
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default     = {}
}
```

outputs.tf:

```
output "aks_id" {
  description = "ID of the AKS cluster"
  value       = azurerm_kubernetes_cluster.aks.id
}

output "aks_fqdn" {
  description = "FQDN of the AKS cluster"
  value       = azurerm_kubernetes_cluster.aks.fqdn
}
```

Step 3: Configure Root Module

main.tf:

```
# Simplified main.tf - Root Module for Student Subscription
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 3.0"
    }
  }
}

provider "azurerm" {
  features {}
}

# Create Resource Group first
resource "azurerm_resource_group" "rg" {
  name     = var.resource_group_name
  location = var.location
  tags     = var.tags
}

# Create VNET
```

```
module "vnet" {
  source              = "./modules/vnet"
  vnet_name           = var.vnet_name
  address_space       = var.vnet_address_space
  location            = var.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags

  depends_on = [azurerm_resource_group.rg]
}

# Create Subnets
module "subnets" {
  source              = "./modules/subnets"
  subnets             = var.subnets
  resource_group_name = azurerm_resource_group.rg.name
  vnet_name           = module.vnet.vnet_name

  depends_on = [module.vnet]
}

# Create NSG with simplified rules
module "nsg" {
  source              = "./modules/nsg"
  nsg_name            = var.nsg_name
  location            = var.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags

  depends_on = [azurerm_resource_group.rg]
}

# Create NAT Gateway
#module "nat_gateway" {
 # source              = "./modules/nat_gateway"
  #nat_gateway_name    = var.nat_gateway_name
  #location            = var.location
  #resource_group_name = azurerm_resource_group.rg.name
  #tags                = var.tags

  #depends_on = [azurerm_resource_group.rg]
#}

# Create Load Balancer
module "load_balancer" {
  source              = "./modules/load_balancer"
  lb_name             = var.lb_name
  location            = var.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags

  depends_on = [azurerm_resource_group.rg]
}

# REMOVED: Route Table module - Not needed with loadBalancer outbound type

# Wait for all core networking resources
resource "time_sleep" "wait_for_core_networking" {
  depends_on = [
    module.vnet,
    module.subnets,
    module.nsg,
    module.load_balancer
  ]
  create_duration = "30s"  # Reduced wait time
}

# NSG Associations
resource "azurerm_subnet_network_security_group_association" "aks_nsg" {
  subnet_id                 = module.subnets.subnet_ids["aks"]
  network_security_group_id = module.nsg.nsg_id

  depends_on = [time_sleep.wait_for_core_networking]
}
```

```
resource "azurerm_subnet_network_security_group_association" "vmss_nsg" {
  subnet_id                 = module.subnets.subnet_ids["vmss"]
  network_security_group_id = module.nsg.nsg_id

  depends_on = [time_sleep.wait_for_core_networking]
}

resource "azurerm_subnet_network_security_group_association" "appgw_nsg" {
  subnet_id                 = module.subnets.subnet_ids["appgateway"]
  network_security_group_id = module.nsg.nsg_id

  depends_on = [time_sleep.wait_for_core_networking]
}

# NAT Gateway Associations
#resource "azurerm_subnet_nat_gateway_association" "aks_nat" {
  #subnet_id      = module.subnets.subnet_ids["aks"]
  #nat_gateway_id = module.nat_gateway.nat_gateway_id

  #depends_on = [time_sleep.wait_for_core_networking]
#}

#resource "azurerm_subnet_nat_gateway_association" "vmss_nat" {
  #subnet_id      = module.subnets.subnet_ids["vmss"]
  #nat_gateway_id = module.nat_gateway.nat_gateway_id

  #depends_on = [time_sleep.wait_for_core_networking]
#}

# Wait for associations
resource "time_sleep" "wait_for_associations" {
  depends_on = [
    azurerm_subnet_network_security_group_association.aks_nsg,
    azurerm_subnet_network_security_group_association.vmss_nsg,
    azurerm_subnet_network_security_group_association.appgw_nsg
    #azurerm_subnet_nat_gateway_association.aks_nat,
    #azurerm_subnet_nat_gateway_association.vmss_nat
  ]
  create_duration = "30s"
}

# Create Application Gateway
module "app_gateway" {
  source              = "./modules/app_gateway"
  app_gateway_name    = var.app_gateway_name
  location            = var.location
  resource_group_name = azurerm_resource_group.rg.name
  subnet_id           = module.subnets.subnet_ids["appgateway"]
  tags                = var.tags

  depends_on = [time_sleep.wait_for_associations]
}

# Create VMSS
module "vmss" {
  source              = "./modules/vmss"
  vmss_name           = var.vmss_name
  location            = var.location
  resource_group_name = azurerm_resource_group.rg.name
  vm_size             = var.vm_size
  instance_count      = var.instance_count
  admin_username      = var.admin_username
  ssh_public_key_path = var.ssh_public_key_path
  subnet_id           = module.subnets.subnet_ids["vmss"]
  backend_pool_id     = module.load_balancer.backend_pool_id
  tags                = var.tags

  depends_on = [time_sleep.wait_for_associations]
}

# Create AKS - Now with simplified networking
module "aks" {
  source              = "./modules/aks"
  aks_name            = var.aks_name
  location            = var.location
  resource_group_name = azurerm_resource_group.rg.name
  kubernetes_version  = var.kubernetes_version
  node_count          = var.node_count
  vm_size             = var.aks_vm_size
  subnet_id           = module.subnets.subnet_ids["aks"]
  tags                = var.tags

  depends_on = [time_sleep.wait_for_associations]
}
```

variables.tf:

```
variable "resource_group_name" {
  description = "Name of the resource group"
  type        = string
  default     = "aks-multi-az-rg"
}

variable "location" {
  description = "Azure region"
  type        = string
  default     = "East US"
}

variable "vnet_name" {
  description = "Name of the virtual network"
  type        = string
  default     = "aks-vnet"
}

variable "vnet_address_space" {
  description = "Address space for the VNET"
  type        = list(string)
  default     = ["10.0.0.0/16"]
}
```

```
variable "subnets" {
  description = "Map of subnet names to address prefixes"
  type        = map(object({
    address_prefix = string
  }))
  default = {
    "aks"        = { address_prefix = "10.0.1.0/24" }
    "appgateway" = { address_prefix = "10.0.2.0/24" }
    "vmss"       = { address_prefix = "10.0.3.0/24" }
  }
}

variable "nsg_name" {
  description = "Name of the NSG"
  type        = string
  default     = "aks-nsg"
}

# REMOVED: Route table variables - not needed with loadBalancer outbound t

variable "nat_gateway_name" {
  description = "Name of the NAT Gateway"
  type        = string
  default     = "aks-nat-gateway"
}

variable "app_gateway_name" {
  description = "Name of the Application Gateway"
  type        = string
  default     = "aks-app-gateway"
}

variable "lb_name" {
  description = "Name of the Load Balancer"
  type        = string
  default     = "aks-load-balancer"
}
```

```
variable "vmss_name" {
  description = "Name of the VMSS"
  type        = string
  default     = "aks-vmss"
}

variable "vm_size" {
  description = "VM size for the scale set"
  type        = string
  default     = "Standard_B1s"
}

variable "instance_count" {
  description = "Number of VM instances"
  type        = number
  default     = 1
}

variable "admin_username" {
  description = "Admin username for VMs"
  type        = string
  default     = "azureuser"
}

variable "ssh_public_key_path" {
  description = "Path to the SSH public key"
  type        = string
  default     = "~/.ssh/id_rsa.pub"
}

variable "aks_name" {
  description = "Name of the AKS cluster"
  type        = string
  default     = "aks-cluster"
}

variable "kubernetes_version" {
  description = "Kubernetes version"
  type        = string
  default     = "1.28"
}

variable "node_count" {
  description = "Number of nodes in the default node pool"
  type        = number
  default     = 1
}

variable "aks_vm_size" {
  description = "VM size for the AKS node pool"
  type        = string
  default     = "Standard_B1s"
}

variable "tags" {
  description = "Tags for resources"
  type        = map(string)
  default = {
    environment = "development"
    project     = "aks-multi-az"
  }
}
```

outputs.tf:

```
output "vnet_id" {
  description = "ID of the VNET"
  value       = module.vnet.vnet_id
}

output "subnet_ids" {
  description = "Map of subnet names to their IDs"
  value       = module.subnets.subnet_ids
}

output "nsg_id" {
  description = "ID of the NSG"
  value       = module.nsg.nsg_id
}

# REMOVED: Route table output - not needed with loadBalancer outbound typ

#output "nat_gateway_id" {
  #description = "ID of the NAT Gateway"
  #value       = module.nat_gateway.nat_gateway_id
#}

output "app_gateway_id" {
  description = "ID of the Application Gateway"
  value       = module.app_gateway.app_gateway_id
}

output "lb_id" {
  description = "ID of the Load Balancer"
  value       = module.load_balancer.lb_id
}

output "vmss_id" {
  description = "ID of the VMSS"
  value       = module.vmss.vmss_id
}

output "aks_id" {
  description = "ID of the AKS cluster"
  value       = module.aks.aks_id
}

output "aks_fqdn" {
  description = "FQDN of the AKS cluster"
  value       = module.aks.aks_fqdn
}
```

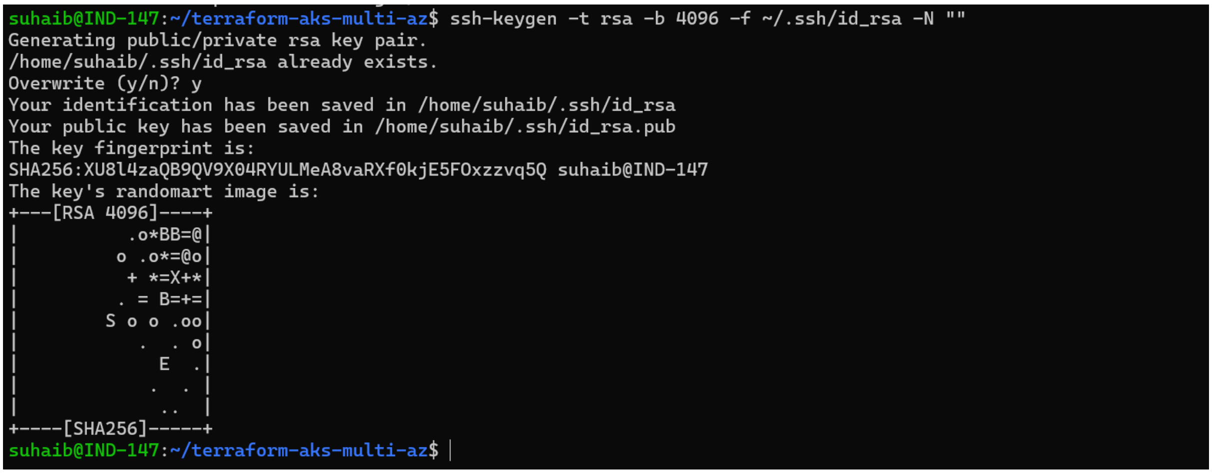
terraform.tfvars:

```
resource_group_name = "aks-multi-az-rg"
location            = "East US"
vnet_name           = "aks-vnet"
vnet_address_space  = ["10.0.0.0/16"]

subnets = {
  "aks"        = { address_prefix = "10.0.1.0/24" }
  "appgateway" = { address_prefix = "10.0.2.0/24" }
  "vmss"       = { address_prefix = "10.0.3.0/24" }
}

nsg_name = "aks-nsg"

nat_gateway_name = "aks-nat-gateway"
app_gateway_name = "aks-app-gateway"
lb_name          = "aks-load-balancer"
vmss_name        = "aks-vmss"

# Student subscription friendly VM sizes
vm_size             = "Standard_B2ms"
instance_count      = 1
admin_username      = "azureuser"
ssh_public_key_path = "~/.ssh/id_rsa.pub"

aks_name           = "aks-cluster"
kubernetes_version = "1.32.4"

node_count         = 1
aks_vm_size        = "Standard_B2ms"

tags = {
  environment = "development"
  project     = "aks-multi-az"
  owner       = "student"
}
```

Phase 3: Deployment and Verification

Step 1: Generate SSH Key Pair

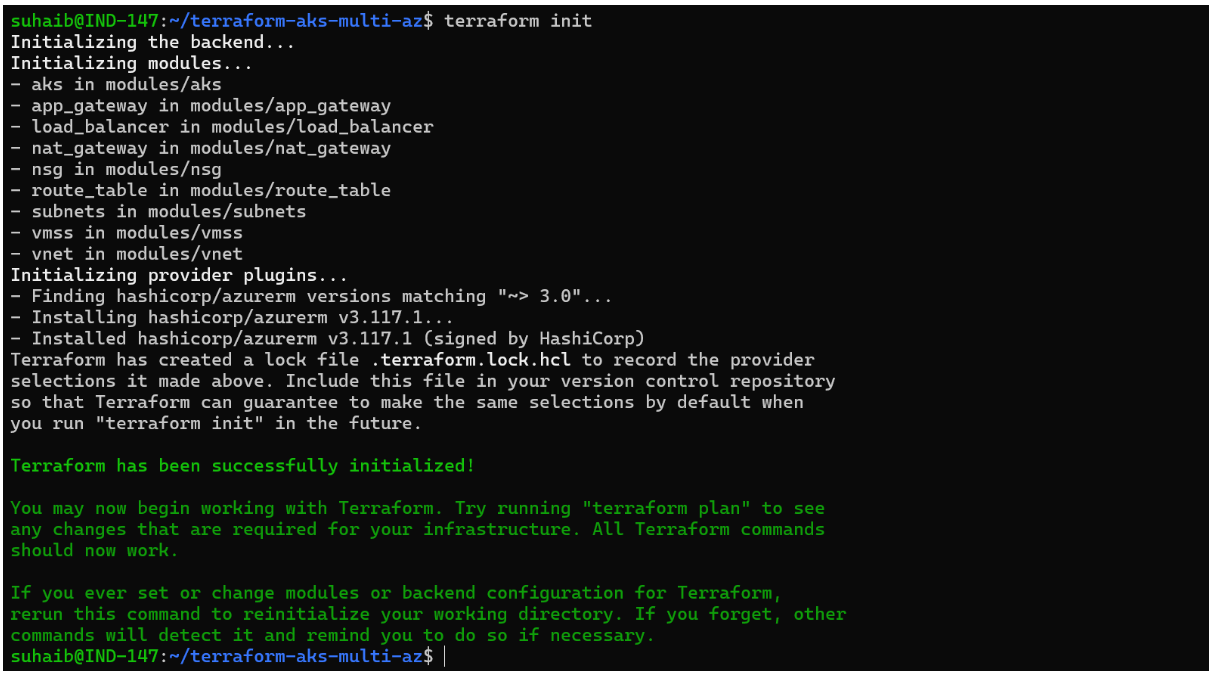
```
# Generate SSH key pair for VMSS and AKS nodes
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa -N ""
```



```
suhaib@IND-147:~/terraform-aks-multi-az$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa -N ""
Generating public/private rsa key pair.
/home/suhaib/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/suhaib/.ssh/id_rsa
Your public key has been saved in /home/suhaib/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:XU8l4zaQB9QV9X04RYULMeA8vaRXf0kjE5FOxzzvq5Q suhaib@IND-147
The key's randomart image is:
+---[RSA 4096]----+
|          .o*BB=@|
|         o .o*=@o|
|          + *=X+*|
|         . = B=+=|
|        S o o .oo|
|           .  . o|
|             E  .|
|            .  . |
|             ..  |
+----[SHA256]-----+
suhaib@IND-147:~/terraform-aks-multi-az$ |
```

Step 2: Initialize Terraform

```
cd ~/terraform-aks-multi-az

# Initialize Terraform
terraform init
```



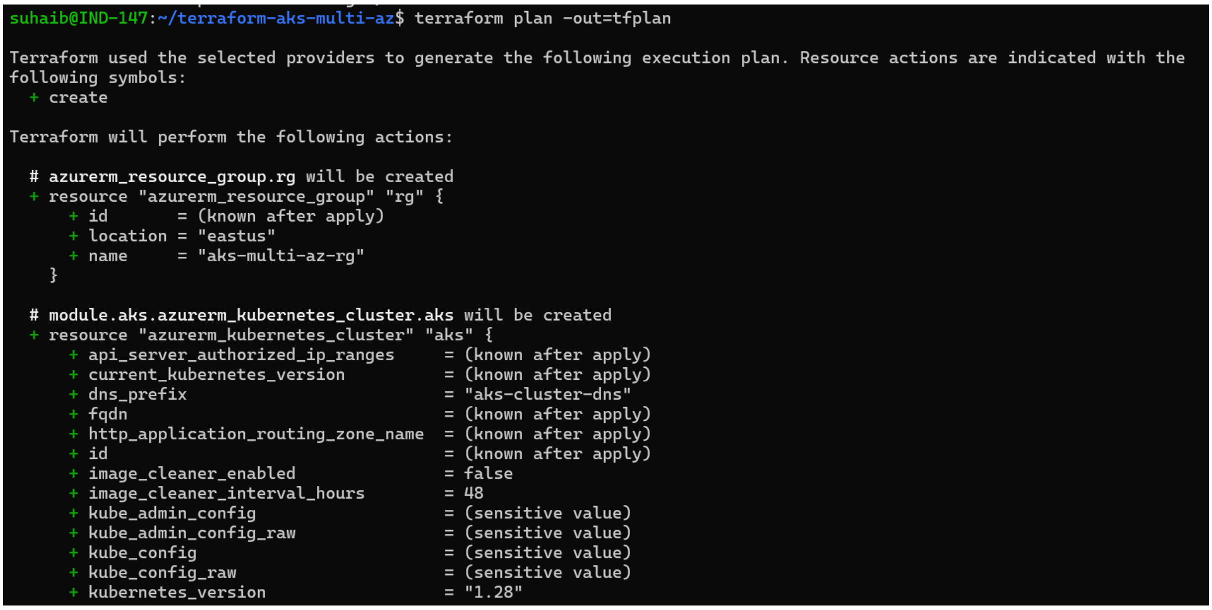
```
suhaib@IND-147:~/terraform-aks-multi-az$ terraform init
Initializing the backend...
Initializing modules...
- aks in modules/aks
- app_gateway in modules/app_gateway
- load_balancer in modules/load_balancer
- nat_gateway in modules/nat_gateway
- nsg in modules/nsg
- route_table in modules/route_table
- subnets in modules/subnets
- vmss in modules/vmss
- vnet in modules/vnet
Initializing provider plugins...
- Finding hashicorp/azurerm versions matching "~> 3.0"...
- Installing hashicorp/azurerm v3.117.1...
- Installed hashicorp/azurerm v3.117.1 (signed by HashiCorp)
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.
suhaib@IND-147:~/terraform-aks-multi-az$ |
```

Expected Output:

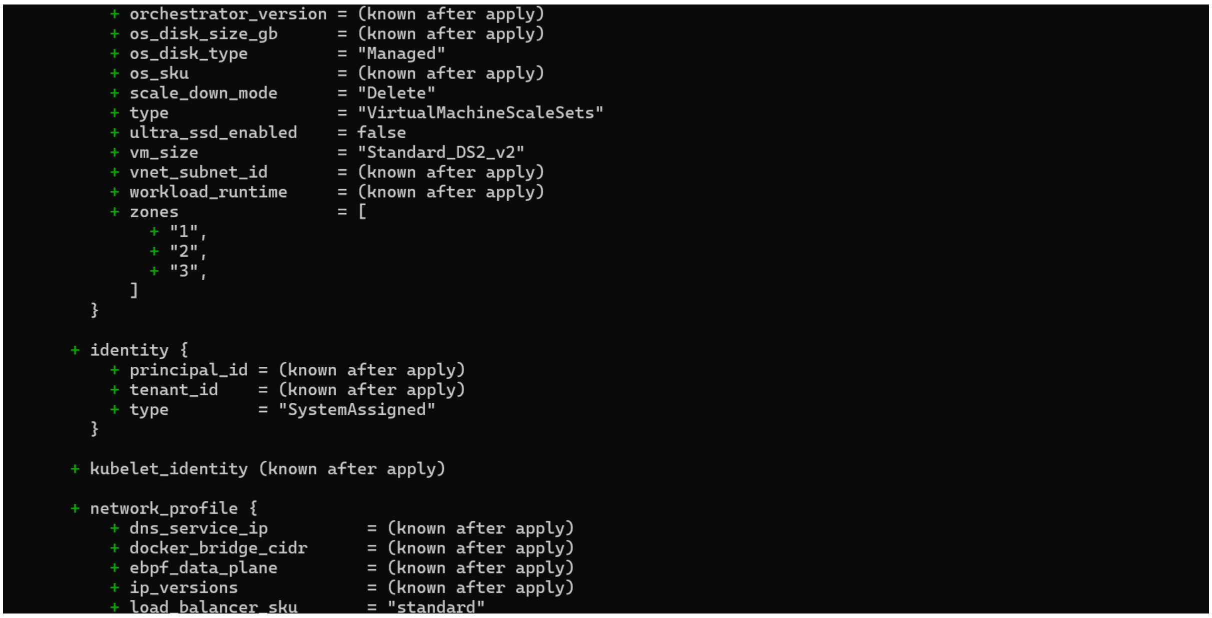
```
Initializing modules...
Initializing the backend...
Initializing provider plugins...
Terraform has been successfully initialized!
```

Step 3: Plan and Apply Configuration

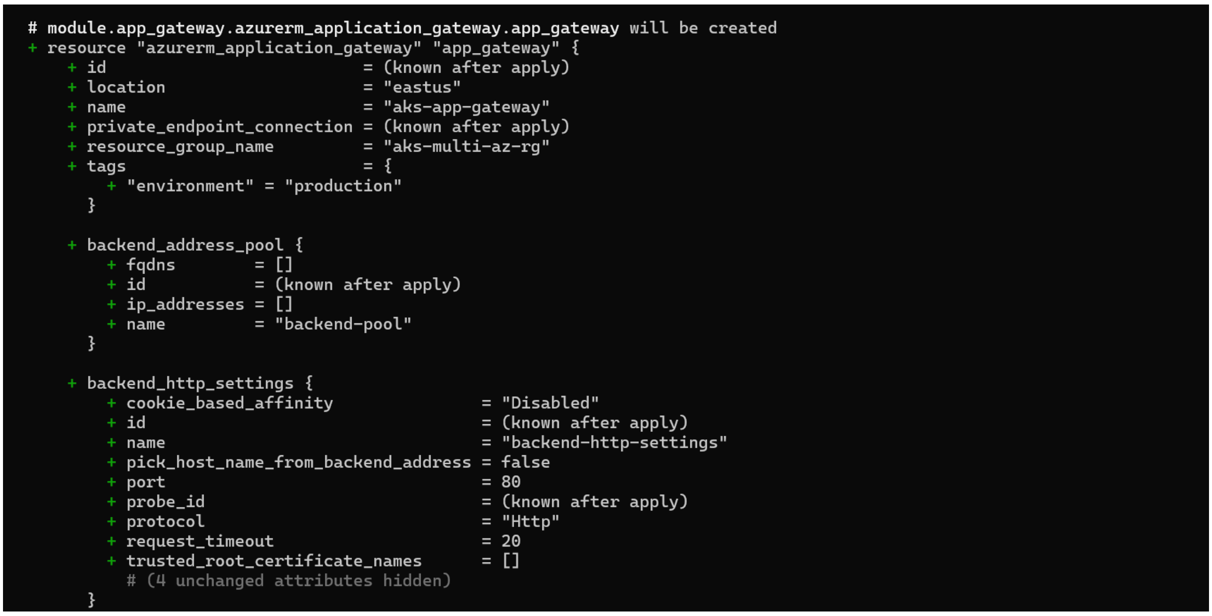
```
# Generate and review execution plan
terraform plan -out=tfplan

# Apply the configuration
terraform apply tfplan
```



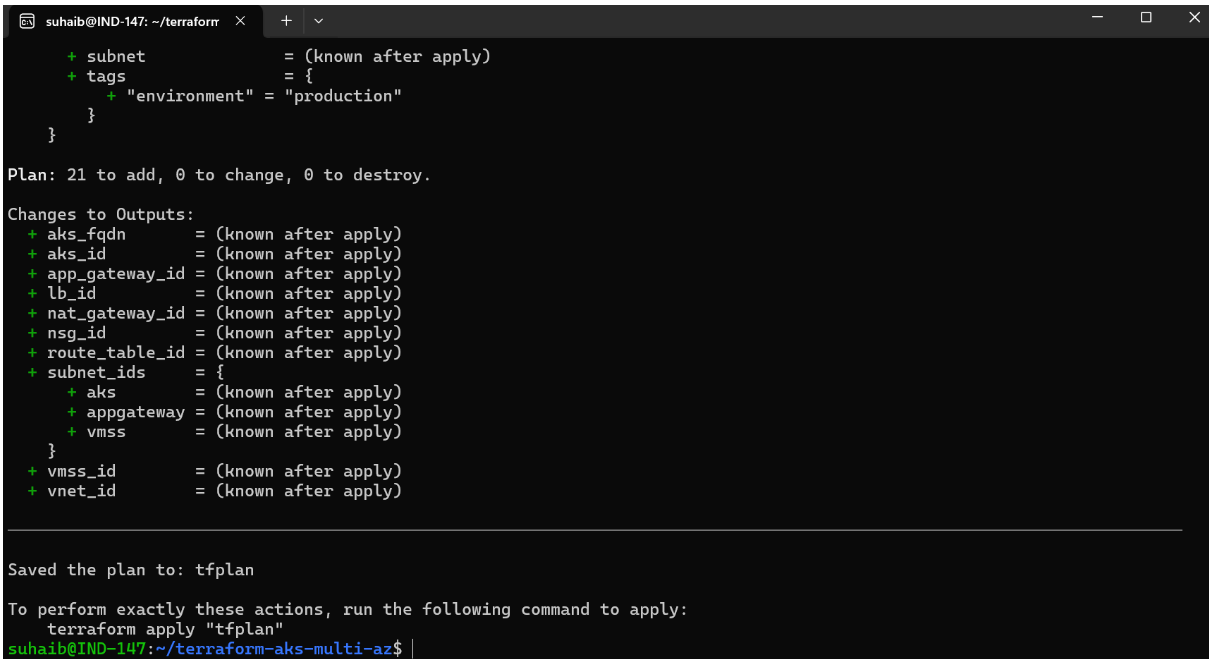
```
suhaib@IND-147:~/terraform-aks-multi-az$ terraform plan -out=tfplan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_resource_group.rg will be created
  + resource "azurerm_resource_group" "rg" {
      + id       = (known after apply)
      + location = "eastus"
      + name     = "aks-multi-az-rg"
    }

  # module.aks.azurerm_kubernetes_cluster.aks will be created
  + resource "azurerm_kubernetes_cluster" "aks" {
      + api_server_authorized_ip_ranges     = (known after apply)
      + current_kubernetes_version          = (known after apply)
      + dns_prefix                          = "aks-cluster-dns"
      + fqdn                                = (known after apply)
      + http_application_routing_zone_name  = (known after apply)
      + id                                  = (known after apply)
      + image_cleaner_enabled               = false
      + image_cleaner_interval_hours        = 48
      + kube_admin_config                   = (sensitive value)
      + kube_admin_config_raw               = (sensitive value)
      + kube_config                         = (sensitive value)
      + kube_config_raw                     = (sensitive value)
      + kubernetes_version                  = "1.28"
```



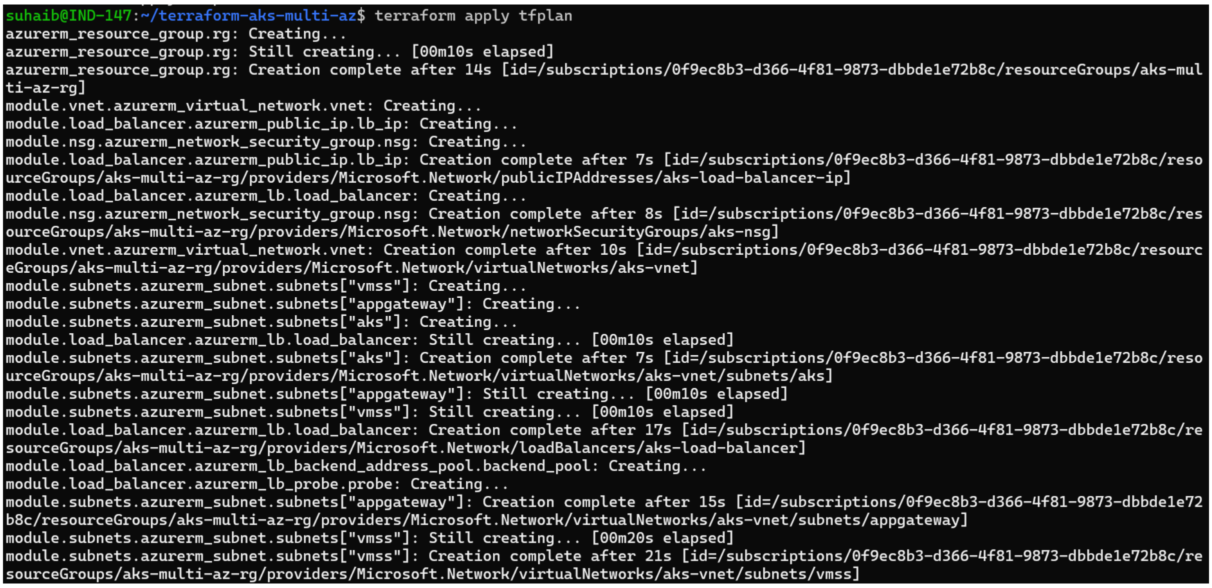
```
          + orchestrator_version = (known after apply)
          + os_disk_size_gb      = (known after apply)
          + os_disk_type         = "Managed"
          + os_sku               = (known after apply)
          + scale_down_mode      = "Delete"
          + type                 = "VirtualMachineScaleSets"
          + ultra_ssd_enabled    = false
          + vm_size              = "Standard_DS2_v2"
          + vnet_subnet_id       = (known after apply)
          + workload_runtime     = (known after apply)
          + zones                = [
              + "1",
              + "2",
              + "3",
            ]
        }

      + identity {
          + principal_id = (known after apply)
          + tenant_id    = (known after apply)
          + type         = "SystemAssigned"
        }

      + kubelet_identity (known after apply)

      + network_profile {
          + dns_service_ip          = (known after apply)
          + docker_bridge_cidr      = (known after apply)
          + ebpf_data_plane         = (known after apply)
          + ip_versions             = (known after apply)
          + load_balancer_sku       = "standard"
```



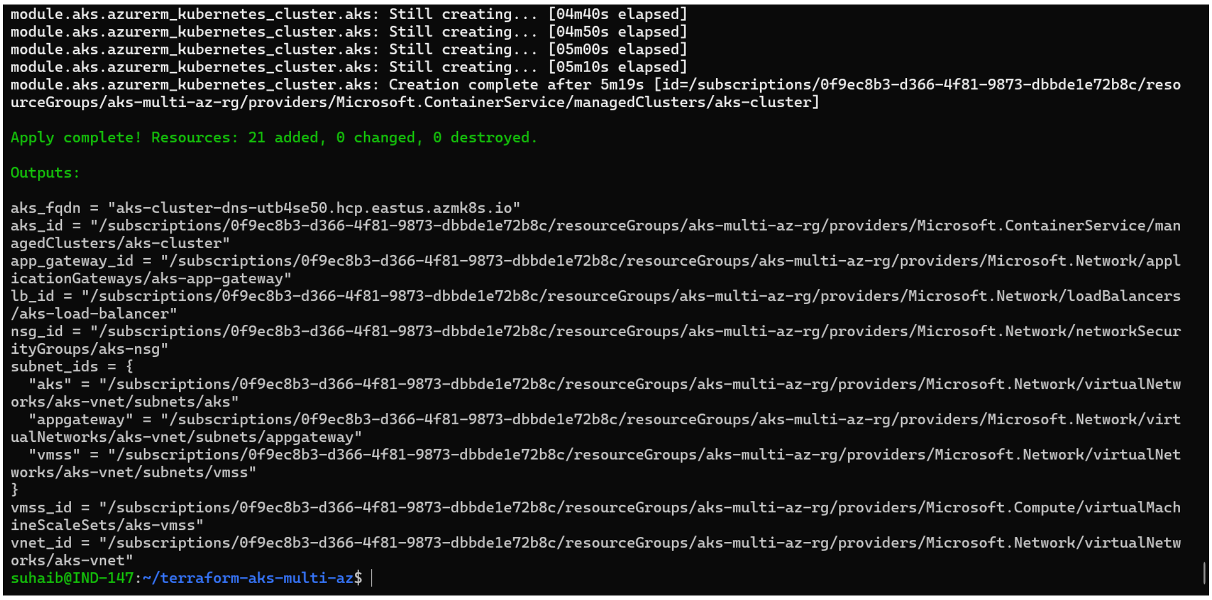
```
  # module.app_gateway.azurerm_application_gateway.app_gateway will be created
  + resource "azurerm_application_gateway" "app_gateway" {
      + id                          = (known after apply)
      + location                    = "eastus"
      + name                        = "aks-app-gateway"
      + private_endpoint_connection = (known after apply)
      + resource_group_name         = "aks-multi-az-rg"
      + tags                        = {
          + "environment" = "production"
        }

      + backend_address_pool {
          + fqdns        = []
          + id           = (known after apply)
          + ip_addresses = []
          + name         = "backend-pool"
        }

      + backend_http_settings {
          + cookie_based_affinity               = "Disabled"
          + id                                  = (known after apply)
          + name                                = "backend-http-settings"
          + pick_host_name_from_backend_address = false
          + port                                = 80
          + probe_id                            = (known after apply)
          + protocol                            = "Http"
          + request_timeout                     = 20
          + trusted_root_certificate_names      = []
            # (4 unchanged attributes hidden)
        }
```



```
      + subnet              = (known after apply)
      + tags                = {
          + "environment" = "production"
        }
    }

Plan: 21 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + aks_fqdn       = (known after apply)
  + aks_id         = (known after apply)
  + app_gateway_id = (known after apply)
  + lb_id          = (known after apply)
  + nat_gateway_id = (known after apply)
  + nsg_id         = (known after apply)
  + route_table_id = (known after apply)
  + subnet_ids     = {
      + aks        = (known after apply)
      + appgateway = (known after apply)
      + vmss       = (known after apply)
    }
  + vmss_id        = (known after apply)
  + vnet_id        = (known after apply)

────────────────────────────────────────────────────────────────────────────────

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
    terraform apply "tfplan"
suhaib@IND-147:~/terraform-aks-multi-az$
```



```
suhaib@IND-147:~/terraform-aks-multi-az$ terraform apply tfplan
azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Still creating... [00m10s elapsed]
azurerm_resource_group.rg: Creation complete after 14s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-mul
ti-az-rg]
module.vnet.azurerm_virtual_network.vnet: Creating...
module.load_balancer.azurerm_public_ip.lb_ip: Creating...
module.nsg.azurerm_network_security_group.nsg: Creating...
module.load_balancer.azurerm_public_ip.lb_ip: Creation complete after 7s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/reso
urceGroups/aks-multi-az-rg/providers/Microsoft.Network/publicIPAddresses/aks-load-balancer-ip]
module.load_balancer.azurerm_lb.load_balancer: Creating...
module.nsg.azurerm_network_security_group.nsg: Creation complete after 8s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/res
ourceGroups/aks-multi-az-rg/providers/Microsoft.Network/networkSecurityGroups/aks-nsg]
module.vnet.azurerm_virtual_network.vnet: Creation complete after 10s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourc
eGroups/aks-multi-az-rg/providers/Microsoft.Network/virtualNetworks/aks-vnet]
module.subnets.azurerm_subnet.subnets["vmss"]: Creating...
module.subnets.azurerm_subnet.subnets["appgateway"]: Creating...
module.subnets.azurerm_subnet.subnets["aks"]: Creating...
module.load_balancer.azurerm_lb.load_balancer: Still creating... [00m10s elapsed]
module.subnets.azurerm_subnet.subnets["aks"]: Creation complete after 7s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/reso
urceGroups/aks-multi-az-rg/providers/Microsoft.Network/virtualNetworks/aks-vnet/subnets/aks]
module.subnets.azurerm_subnet.subnets["appgateway"]: Still creating... [00m10s elapsed]
module.subnets.azurerm_subnet.subnets["vmss"]: Still creating... [00m10s elapsed]
module.load_balancer.azurerm_lb.load_balancer: Creation complete after 17s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/re
sourceGroups/aks-multi-az-rg/providers/Microsoft.Network/loadBalancers/aks-load-balancer]
module.load_balancer.azurerm_lb_backend_address_pool.backend_pool: Creating...
module.load_balancer.azurerm_lb_probe.probe: Creating...
module.subnets.azurerm_subnet.subnets["appgateway"]: Creation complete after 15s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72
b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/virtualNetworks/aks-vnet/subnets/appgateway]
module.subnets.azurerm_subnet.subnets["vmss"]: Still creating... [00m20s elapsed]
module.subnets.azurerm_subnet.subnets["vmss"]: Creation complete after 21s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/re
sourceGroups/aks-multi-az-rg/providers/Microsoft.Network/virtualNetworks/aks-vnet/subnets/vmss]
```



```
module.aks.azurerm_kubernetes_cluster.aks: Still creating... [04m40s elapsed]
module.aks.azurerm_kubernetes_cluster.aks: Still creating... [04m50s elapsed]
module.aks.azurerm_kubernetes_cluster.aks: Still creating... [05m00s elapsed]
module.aks.azurerm_kubernetes_cluster.aks: Still creating... [05m10s elapsed]
module.aks.azurerm_kubernetes_cluster.aks: Creation complete after 5m19s [id=/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/reso
urceGroups/aks-multi-az-rg/providers/Microsoft.ContainerService/managedClusters/aks-cluster]

Apply complete! Resources: 21 added, 0 changed, 0 destroyed.

Outputs:

aks_fqdn = "aks-cluster-dns-utb4se50.hcp.eastus.azmk8s.io"
aks_id = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.ContainerService/man
agedClusters/aks-cluster"
app_gateway_id = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/appl
icationGateways/aks-app-gateway"
lb_id = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/loadBalancers
/aks-load-balancer"
nsg_id = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/networkSecur
ityGroups/aks-nsg"
subnet_ids = {
  "aks" = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/virtualNetw
orks/aks-vnet/subnets/aks"
  "appgateway" = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/virt
ualNetworks/aks-vnet/subnets/appgateway"
  "vmss" = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/virtualNet
works/aks-vnet/subnets/vmss"
}
vmss_id = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Compute/virtualMach
ineScaleSets/aks-vmss"
vnet_id = "/subscriptions/0f9ec8b3-d366-4f81-9873-dbbde1e72b8c/resourceGroups/aks-multi-az-rg/providers/Microsoft.Network/virtualNetw
orks/aks-vnet"
suhaib@IND-147:~/terraform-aks-multi-az$ |
```

Expected Duration: 10-15 minutes

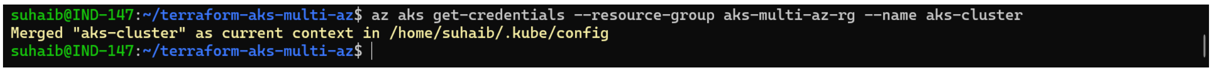
Key Resources Created:

- Resource Group: `aks-multi-az-rg`
- Virtual Network: `aks-vnet`
- Subnets: AKS, Application Gateway, VMSS
- Network Security Group with appropriate rules
- Application Gateway with public IP
- Load Balancer with backend pool
- Virtual Machine Scale Set

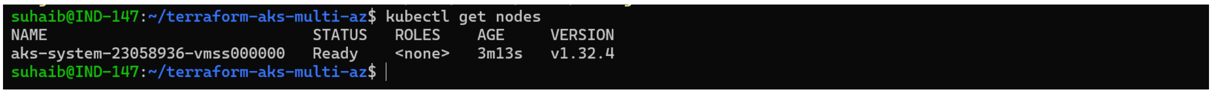
- AKS Cluster with Log Analytics integration

Step 4: Verify AKS Cluster

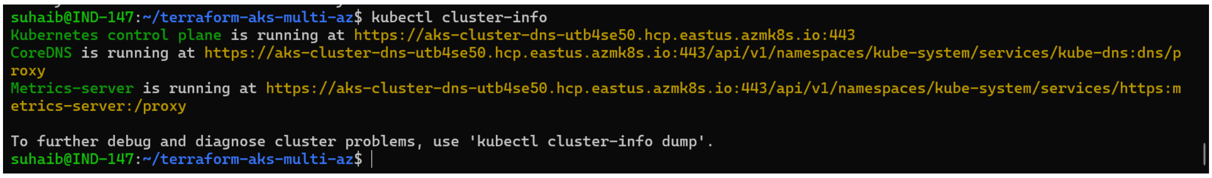
```
# Get AKS credentials
az aks get-credentials \
  --resource-group aks-multi-az-rg \
  --name aks-cluster

# Verify nodes are ready
kubectl get nodes

# Check cluster information
kubectl cluster-info

# Verify cluster health
kubectl get pods --all-namespaces
```



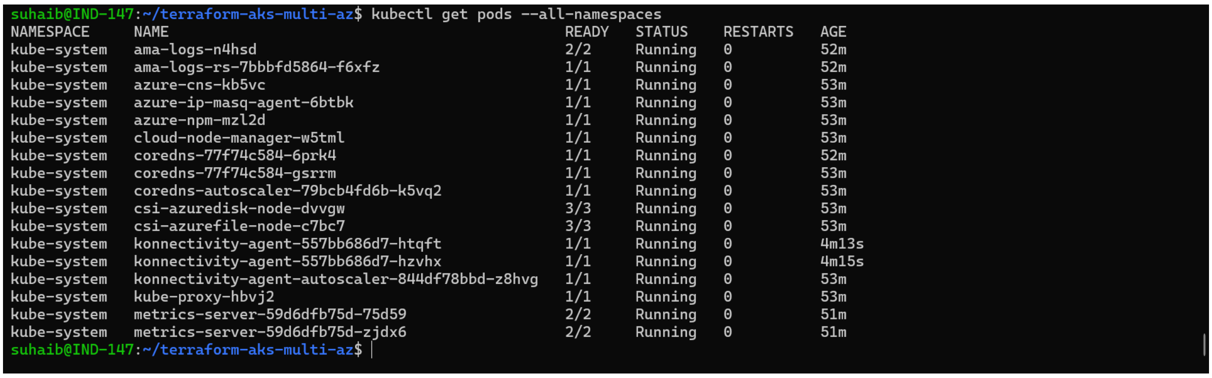
```
suhaib@IND-147:~/terraform-aks-multi-az$ az aks get-credentials --resource-group aks-multi-az-rg --name aks-cluster
Merged "aks-cluster" as current context in /home/suhaib/.kube/config
suhaib@IND-147:~/terraform-aks-multi-az$ 
```



```
suhaib@IND-147:~/terraform-aks-multi-az$ kubectl get nodes
NAME                              STATUS   ROLES    AGE     VERSION
aks-system-23058936-vmss000000    Ready    <none>   3m13s   v1.32.4
suhaib@IND-147:~/terraform-aks-multi-az$ 
```



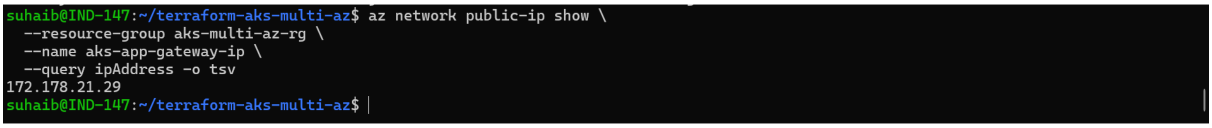
```
suhaib@IND-147:~/terraform-aks-multi-az$ kubectl cluster-info
Kubernetes control plane is running at https://aks-cluster-dns-utb4se50.hcp.eastus.azmk8s.io:443
CoreDNS is running at https://aks-cluster-dns-utb4se50.hcp.eastus.azmk8s.io:443/api/v1/namespaces/kube-system/services/kube-dns:dns/p
roxy
Metrics-server is running at https://aks-cluster-dns-utb4se50.hcp.eastus.azmk8s.io:443/api/v1/namespaces/kube-system/services/https:m
etrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
suhaib@IND-147:~/terraform-aks-multi-az$ 
```



```
suhaib@IND-147:~/terraform-aks-multi-az$ kubectl get pods --all-namespaces
NAMESPACE     NAME                                             READY   STATUS    RESTARTS   AGE
kube-system   ama-logs-n4hsd                                   2/2     Running   0          52m
kube-system   ama-logs-rs-7bbbfd5864-f6xfz                     1/1     Running   0          52m
kube-system   azure-cns-kb5vc                                  1/1     Running   0          53m
kube-system   azure-ip-masq-agent-6btbk                        1/1     Running   0          53m
kube-system   azure-npm-mzl2d                                  1/1     Running   0          53m
kube-system   cloud-node-manager-w5tml                         1/1     Running   0          53m
kube-system   coredns-77f74c584-6prk4                          1/1     Running   0          52m
kube-system   coredns-77f74c584-gsrrm                          1/1     Running   0          53m
kube-system   coredns-autoscaler-79bcb4fd6b-k5vq2              1/1     Running   0          53m
kube-system   csi-azuredisk-node-dvvgw                         3/3     Running   0          53m
kube-system   csi-azurefile-node-c7bc7                         3/3     Running   0          53m
kube-system   konnectivity-agent-557bb686d7-htqft              1/1     Running   0          4m13s
kube-system   konnectivity-agent-557bb686d7-hzvhx              1/1     Running   0          4m15s
kube-system   konnectivity-agent-autoscaler-844df78bbd-z8hvg   1/1     Running   0          53m
kube-system   kube-proxy-hbvj2                                 1/1     Running   0          53m
kube-system   metrics-server-59d6dfb75d-75d59                  2/2     Running   0          51m
kube-system   metrics-server-59d6dfb75d-zjdx6                  2/2     Running   0          51m
suhaib@IND-147:~/terraform-aks-multi-az$ 
```

Step 5: Test Application Gateway

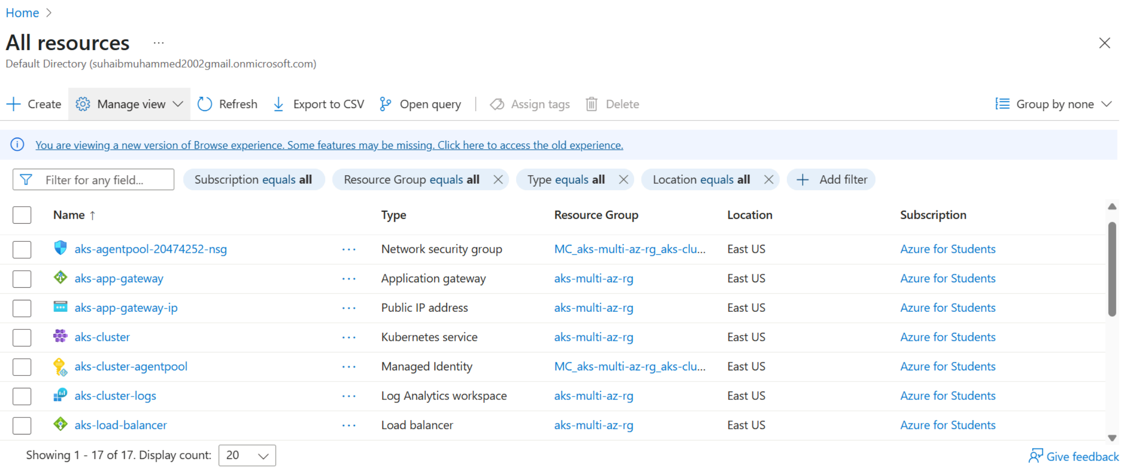
```
# Get Application Gateway public IP
az network public-ip show \
  --resource-group aks-multi-az-rg \
  --name aks-app-gateway-ip \
  --query ipAddress -o tsv
```



```
suhaib@IND-147:~/terraform-aks-multi-az$ az network public-ip show \
  --resource-group aks-multi-az-rg \
  --name aks-app-gateway-ip \
  --query ipAddress -o tsv
172.178.21.29
suhaib@IND-147:~/terraform-aks-multi-az$ 
```

Step 6: Verify Resources in Azure Portal

1. **Login to Azure Portal**: https://portal.azure.com
2. **Navigate to Resource Groups** → `aks-multi-az-rg`
3. **Verify Resources**:
   - Virtual Network: `aks-vnet`
   - Subnets: 3 subnets created
   - Network Security Group: `aks-nsg`
   - Application Gateway: `aks-app-gateway`
   - Load Balancer: `aks-load-balancer`
   - Virtual Machine Scale Set: `aks-vmss`
   - Kubernetes Service: `aks-cluster`

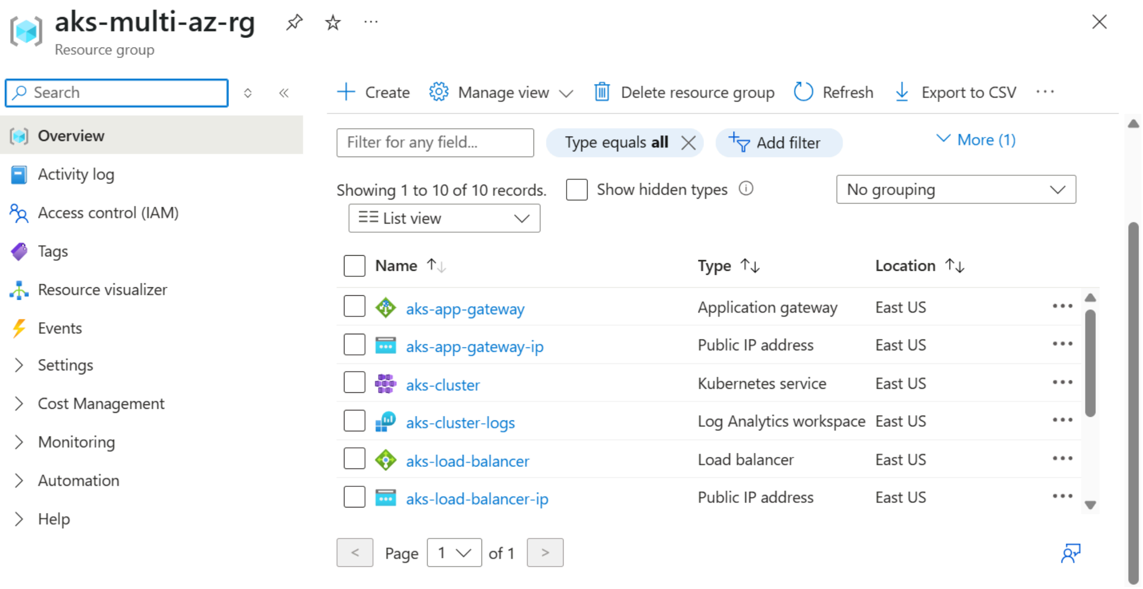


Default Directory (suhaibmuhammed2002gmail.onmicrosoft.com)

You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

□	Name ↑		Type	Resource Group	Location	Subscription
□	aks-agentpool-20474252-nsg	···	Network security group	MC_aks-multi-az-rg_aks-clu...	East US	Azure for Students
□	aks-app-gateway	···	Application gateway	aks-multi-az-rg	East US	Azure for Students
□	aks-app-gateway-ip	···	Public IP address	aks-multi-az-rg	East US	Azure for Students
□	aks-cluster	···	Kubernetes service	aks-multi-az-rg	East US	Azure for Students
□	aks-cluster-agentpool	···	Managed Identity	MC_aks-multi-az-rg_aks-clu...	East US	Azure for Students
□	aks-cluster-logs	···	Log Analytics workspace	aks-multi-az-rg	East US	Azure for Students
□	aks-load-balancer	···	Load balancer	aks-multi-az-rg	East US	Azure for Students

Showing 1 - 17 of 17. Display count: 20



aks-multi-az-rg

Resource group

Search

Overview

Activity log

Access control (IAM)

Tags

Resource visualizer

Events

Settings

Cost Management

Monitoring

Automation

Help

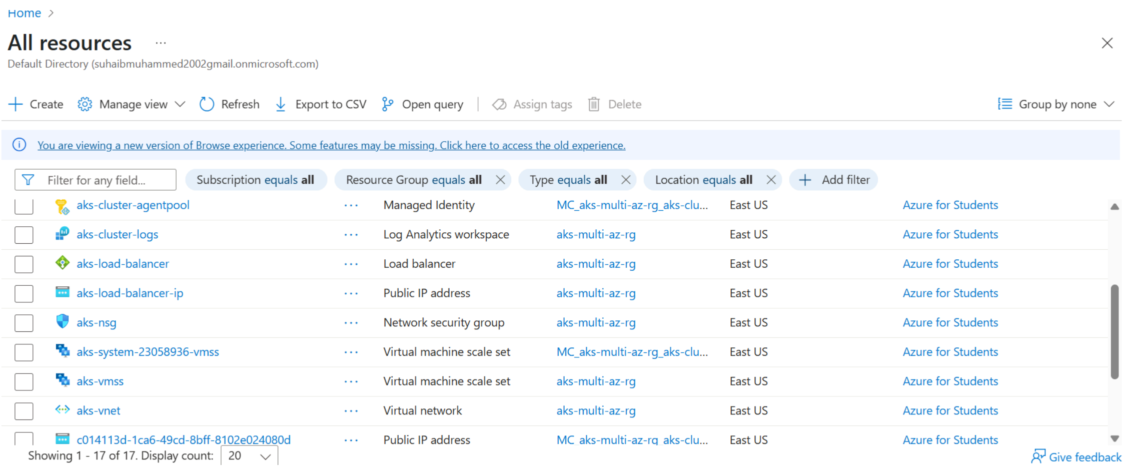
Create | Manage view | Delete resource group | Refresh | Export to CSV

Filter for any field... | Type equals **all** | Add filter | More (1)

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

□	Name	Type	Location
□	aks-app-gateway	Application gateway	East US
□	aks-app-gateway-ip	Public IP address	East US
□	aks-cluster	Kubernetes service	East US
□	aks-cluster-logs	Log Analytics workspace	East US
□	aks-load-balancer	Load balancer	East US
□	aks-load-balancer-ip	Public IP address	East US

< Page 1 of 1 >



Home >

All resources

Default Directory (suhaibmuhammed2002gmail.onmicrosoft.com)

Create | Manage view | Refresh | Export to CSV | Open query | Assign tags | Delete | Group by none

You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

Filter for any field... | Subscription equals **all** | Resource Group equals **all** | Type equals **all** | Location equals **all** | Add filter

□	Name	Type	Resource group	Location	Subscription
□	aks-cluster-agentpool	Managed Identity	MC_aks-multi-az-rg_aks-clu...	East US	Azure for Students
□	aks-cluster-logs	Log Analytics workspace	aks-multi-az-rg	East US	Azure for Students
□	aks-load-balancer	Load balancer	aks-multi-az-rg	East US	Azure for Students
□	aks-load-balancer-ip	Public IP address	aks-multi-az-rg	East US	Azure for Students
□	aks-nsg	Network security group	aks-multi-az-rg	East US	Azure for Students
□	aks-system-23058936-vmss	Virtual machine scale set	MC_aks-multi-az-rg_aks-clu...	East US	Azure for Students
□	aks-vmss	Virtual machine scale set	aks-multi-az-rg	East US	Azure for Students
□	aks-vnet	Virtual network	aks-multi-az-rg	East US	Azure for Students
□	c014113d-1ca6-49cd-8bff-8102e024080d	Public IP address	MC_aks-multi-az-rg_aks-clu...	East US	Azure for Students

Showing 1 - 17 of 17. Display count: 20

Give feedback

Module Documentation

Available Modules

1. VNET Module

- **Path**: `modules/vnet`
- **Purpose**: Creates virtual network with specified address space
- **Inputs**: `vnet_name`, `address_space`, `location`, `resource_group_name`, `tags`
- **Outputs**: `vnet_id`, `vnet_name`

2. Subnets Module

- **Path**: `modules/subnets`

- **Purpose**: Creates multiple subnets within a virtual network
- **Inputs**: `subnets`, `resource_group_name`, `vnet_name`
- **Outputs**: `subnet_ids`

3. NSG Module

- **Path**: `modules/nsg`
- **Purpose**: Creates network security group with predefined rules
- **Inputs**: `nsg_name`, `location`, `resource_group_name`, `tags`
- **Outputs**: `nsg_id`

4. AKS Module

- **Path**: `modules/aks`
- **Purpose**: Creates AKS cluster with monitoring and RBAC enabled
- **Inputs**: `aks_name`, `location`, `resource_group_name`, `kubernetes_version`, `node_count`, `vm_size`, `subnet_id`, `tags`
- **Outputs**: `aks_id`, `aks_fqdn`

Usage Example

```
module "aks" {
  source              = "./modules/aks"
  aks_name            = "my-aks-cluster"
  location            = "East US"
  resource_group_name = "my-resource-group"
  kubernetes_version  = "1.32.4"
  node_count          = 3
  vm_size             = "Standard_B2ms"
  subnet_id           = module.subnets.subnet_ids["aks"]
  tags = {
    environment = "production"
    project     = "my-project"
  }
}
```

Troubleshooting

Common Issues and Solutions

1. Authentication Errors

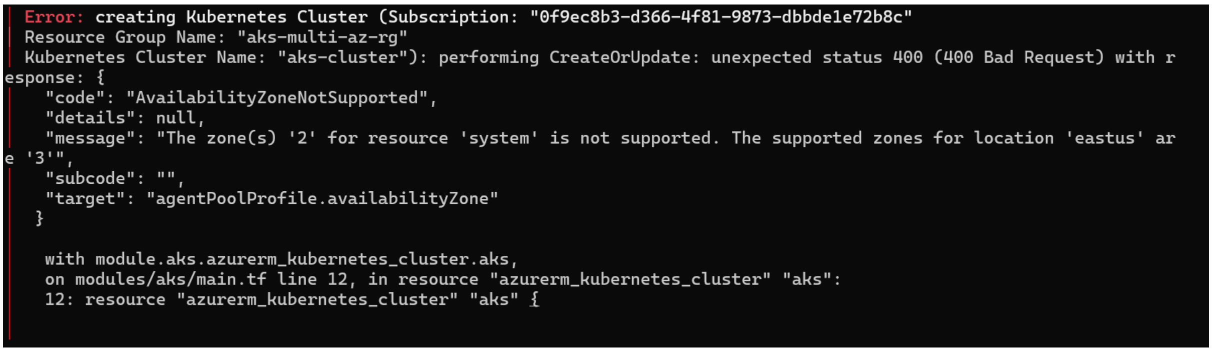
Issue: `Error: Unable to list provider registration status` **Solution**:

```
# Verify environment variables are set
echo $ARM_SUBSCRIPTION_ID
echo $ARM_CLIENT_ID

# Re-authenticate if needed
az login
```

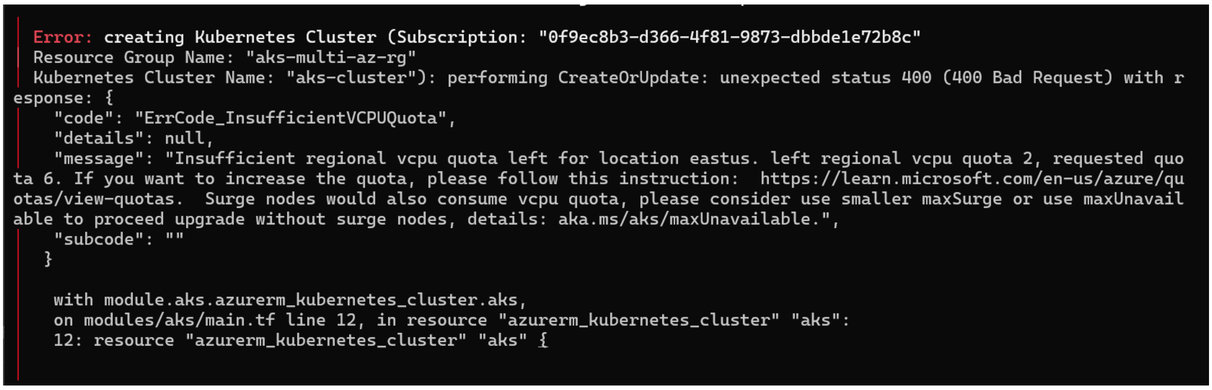
2. Resource Quota Exceeded

Issue: `Quota exceeded for VM family` **Solution**:

- Use smaller VM sizes (e.g., `Standard_B1s` instead of `Standard_B2ms`)
- Request quota increase in Azure Portal
- Choose different Azure region



```
 Error: creating Kubernetes Cluster (Subscription: "0f9ec8b3-d366-4f81-9873-dbbde1e72b8c"
 Resource Group Name: "aks-multi-az-rg"
 Kubernetes Cluster Name: "aks-cluster"): performing CreateOrUpdate: unexpected status 400 (400 Bad Request) with r
esponse: {
    "code": "AvailabilityZoneNotSupported",
    "details": null,
    "message": "The zone(s) '2' for resource 'system' is not supported. The supported zones for location 'eastus' ar
e '3'",
    "subcode": "",
    "target": "agentPoolProfile.availabilityZone"
   }

    with module.aks.azurerm_kubernetes_cluster.aks,
    on modules/aks/main.tf line 12, in resource "azurerm_kubernetes_cluster" "aks":
    12: resource "azurerm_kubernetes_cluster" "aks" {
```



```
 Error: creating Kubernetes Cluster (Subscription: "0f9ec8b3-d366-4f81-9873-dbbde1e72b8c"
 Resource Group Name: "aks-multi-az-rg"
 Kubernetes Cluster Name: "aks-cluster"): performing CreateOrUpdate: unexpected status 400 (400 Bad Request) with r
esponse: {
    "code": "ErrCode_InsufficientVCPUQuota",
    "details": null,
    "message": "Insufficient regional vcpu quota left for location eastus. left regional vcpu quota 2, requested quo
ta 6. If you want to increase the quota, please follow this instruction:  https://learn.microsoft.com/en-us/azure/qu
otas/view-quotas.  Surge nodes would also consume vcpu quota, please consider use smaller maxSurge or use maxUnavail
able to proceed upgrade without surge nodes, details: aka.ms/aks/maxUnavailable.",
    "subcode": ""
   }

    with module.aks.azurerm_kubernetes_cluster.aks,
    on modules/aks/main.tf line 12, in resource "azurerm_kubernetes_cluster" "aks":
    12: resource "azurerm_kubernetes_cluster" "aks" {
```

3. Network Security Group Rules

Issue: Application Gateway health probe failures

Solution: Ensure NSG includes management port rules (65200-65535)

4. Terraform State Issues

Issue: `Resource already exists`**Solution**:

```
# Import existing resource
terraform import azurerm_resource_group.rg /subscriptions/SUB_ID/resourceGroups/RESOURCE_GROUP_NAME

# Or destroy and recreate
terraform destroy
terraform apply
```

Debug Commands

```
# Check Terraform state
terraform state list

# Show specific resource
terraform state show azurerm_kubernetes_cluster.aks

# Validate configuration
terraform validate

# Format configuration files
terraform fmt -recursive
```

Cleanup

Complete Infrastructure Cleanup

```
# Destroy all resources
terraform destroy
```

```
# Confirm with 'yes' when prompted
```

Selective Resource Cleanup

```
# Remove specific resource
terraform destroy -target=module.vmss

# Remove multiple resources
terraform destroy -target=module.vmss -target=module.app_gateway
```

Manual Cleanup (if needed)

```
# Delete resource group (removes all contained resources)
az group delete --name aks-multi-az-rg --yes --no-wait
```

Cost Optimization Tips

For Student Subscriptions

1. **Use Smaller VM Sizes**:
   - AKS: `Standard_B1s` or `Standard_B2s`
   - VMSS: `Standard_B1s`
2. **Reduce Instance Counts**:
   - AKS: 1 node instead of 3
   - VMSS: 1 instance instead of 3
3. **Disable Auto-scaling**:

   ```
   enable_auto_scaling = false
   ```

4. **Use Spot Instances** (for non-production):

   ```
   priority = "Spot"
   eviction_policy = "Deallocate"
   ```

Monitoring Costs

```
# Check current spending
az consumption usage list --output table

# Set up budget alerts in Azure Portal
# Navigate to: Cost Management + Billing > Budgets
```

Security Considerations

Production Hardening

1. **Network Security**:
   - Implement Zero Trust networking
   - Use Azure Firewall for egress control
   - Enable Azure Policy for compliance
2. **Identity and Access**:
   - Use Azure AD integration
   - Implement Pod Identity
   - Enable audit logging
3. **Secrets Management**:
   - Use Azure Key Vault for secrets
   - Implement CSI driver for secret mounting
   - Rotate credentials regularly

Example Security Enhancements

```
# Enable private cluster
private_cluster_enabled = true

# Enable Azure Policy
azure_policy_enabled = true

# Enable secret store CSI driver
```

```
key_vault_secrets_provider {
  secret_rotation_enabled = true
}
```

Support and Resources

Documentation Links

- Azure Kubernetes Service Documentation
- Terraform Azure Provider
- Azure Architecture Center

Community Resources

- Azure Kubernetes Service GitHub
- Terraform Azure Examples

Getting Help

- Azure Support Portal
- Stack Overflow with tags: `azure-aks`, `terraform`, `azure`
- GitHub Issues for specific tools