

Multimodal Reasoning AI Agent

Upload an image and provide a reasoning-based task for the AI Agent. The AI Agent will analyze the image and respond based on your input.

Upload Image



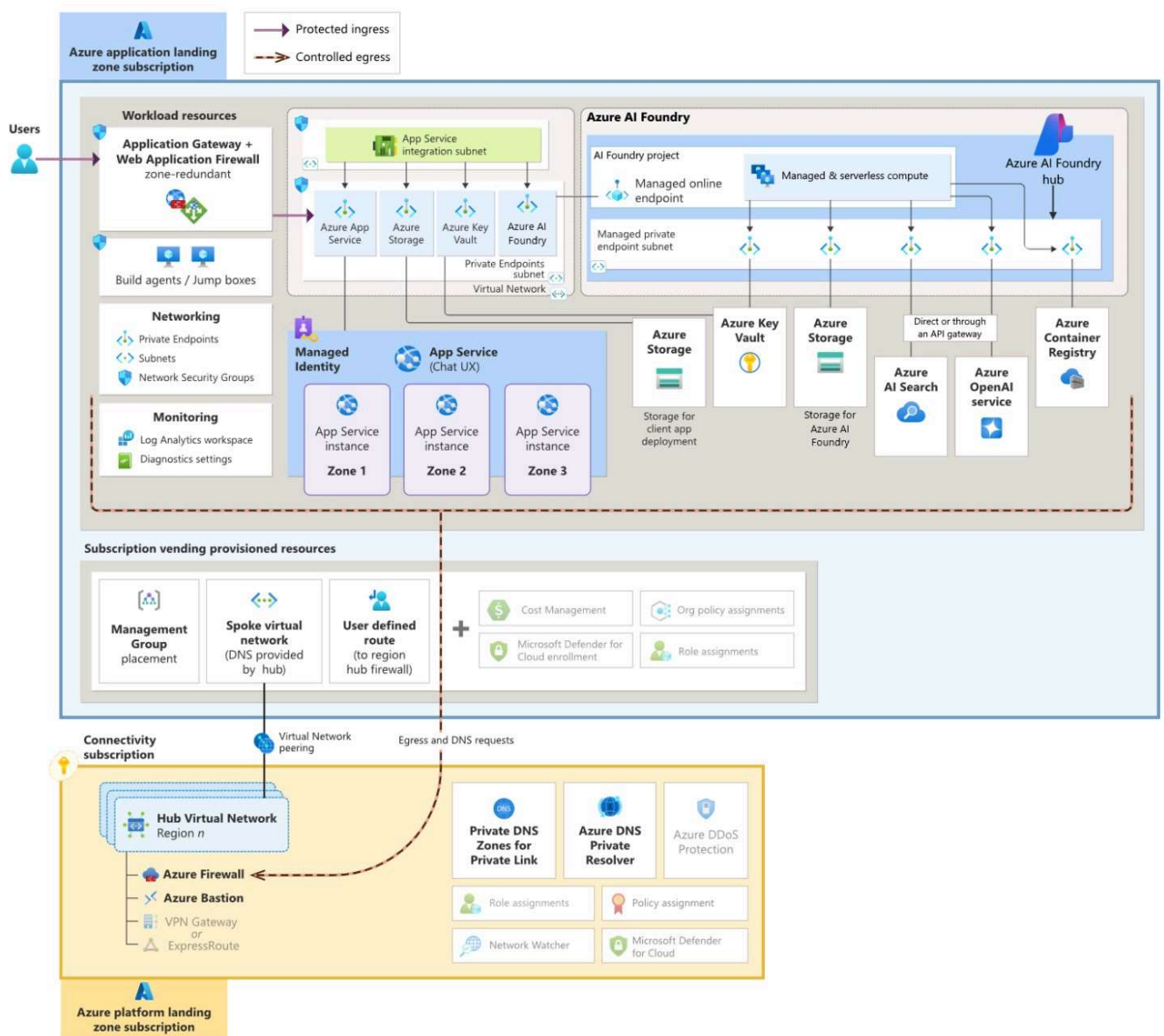
Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



azure-openai-baseline-landing-zone.png 216.5KB



Uploaded Image

Enter your task/question for the AI Agent:

generate code for this architecture

Analyze Image

AI Response:

Azure Architecture Deployment with Terraform

This code provides a basic Terraform configuration to deploy the Azure architecture.

Note: This is a foundational example and might require adjustments based on specific requirements.

Key Components Covered:

- Resource Groups:** Organizing resources.
- Virtual Networks (VNet):** Hub and Spoke network topology.
- Subnets:** Dividing the VNet.
- Network Security Groups (NSGs):** Controlling network traffic.
- Azure Firewall:** Network security appliance.
- Azure Bastion:** Securely connecting to VMs.
- Application Gateway with WAF:** Load balancer with web application firewall.
- App Service:** Hosting web applications.
- Azure Storage:** General-purpose storage.
- Azure Key Vault:** Managing secrets and keys.
- Azure AI Foundry (Conceptual):** Includes managed online endpoint and serverless inference.
- Azure Container Registry (ACR):** Storing container images.
- Azure AI Search:** Search-as-a-service.
- Azure OpenAI Service:** Accessing OpenAI models.
- Private Endpoints:** Securely accessing PaaS services.
- Private DNS Zones and Resolver:** Managing private DNS records.
- Log Analytics Workspace:** Centralized logging.

Files:

- main.tf:** Contains the core resource definitions.
- variables.tf:** Defines input variables.
- outputs.tf:** Displays output values.

main.tf:

```
terraform
terraform {
  required_providers {
    azurerm = {
```

```

        source = "hashicorp/azurerm"
        version = "~> 3.0"
    }
}

provider "azurerm" {
    features {}
}

# Variables (defined in variables.tf)
variable "location" {
    description = "The Azure region to deploy resources into."
    default     = "eastus2"
}

variable "resource_group_name_hub" {
    description = "Name of the resource group for the Hub network."
    default     = "rg-hub-network"
}

variable "resource_group_name_spoke" {
    description = "Name of the resource group for the Spoke network and workload res
    default     = "rg-spoke-workload"
}

variable "hub_vnet_name" {
    description = "Name of the Hub Virtual Network."
    default     = "hub-vnet"
}

variable "hub_vnet_address_space" {
    description = "Address space for the Hub Virtual Network."
    default     = ["10.10.0.0/16"]
}

variable "firewall_subnet_prefix" {
    description = "Address prefix for the Azure Firewall subnet."
    default     = "10.10.1.0/24"
}

variable "bastion_subnet_prefix" {
    description = "Address prefix for the Azure Bastion subnet."
    default     = "10.10.2.0/24"
}

variable "spoke_vnet_name" {
    description = "Name of the Spoke Virtual Network."
    default     = "spoke-vnet"
}

```

```

}

variable "spoke_vnet_address_space" {
  description = "Address space for the Spoke Virtual Network."
  default     = ["10.20.0.0/16"]
}

variable "app_gateway_subnet_prefix" {
  description = "Address prefix for the Application Gateway subnet."
  default     = "10.20.1.0/24"
}

variable "app_service_subnet_prefix" {
  description = "Address prefix for the App Service integration subnet."
  default     = "10.20.2.0/24"
}

variable "private_endpoints_subnet_prefix" {
  description = "Address prefix for the Private Endpoints subnet."
  default     = "10.20.3.0/24"
}

variable "storage_account_tier" {
  description = "Tier for the Storage Account."
  default     = "Standard"
}

variable "storage_account_replication_type" {
  description = "Replication type for the Storage Account."
  default     = "LRS"
}

variable "app_service_plan_sku_name" {
  description = "SKU name for the App Service Plan."
  default     = "P1v2"
}

variable "app_service_plan_capacity" {
  description = "Capacity for the App Service Plan."
  default     = 3
}

# Resource Group (Hub)
resource "azurerm_resource_group" "rg_hub" {
  name       = var.resource_group_name_hub
  location   = var.location
}

```

```

# Resource Group (Spoke)

```

```
resource "azurerm_resource_group" "rg_spoke" {
  name      = var.resource_group_name_spoke
  location  = var.location
}
```

Hub Virtual Network

```
resource "azurerm_virtual_network" "hub_vnet" {
  name                = var.hub_vnet_name
  location            = azurerm_resource_group.rg_hub.location
  resource_group_name = azurerm_resource_group.rg_hub.name
  address_space       = var.hub_vnet_address_space
}
```

Hub Subnets

```
resource "azurerm_subnet" "firewall_subnet" {
  name                = "AzureFirewallSubnet"
  resource_group_name = azurerm_resource_group.rg_hub.name
  virtual_network_name = azurerm_virtual_network.hub_vnet.name
  address_prefixes    = [var.firewall_subnet_prefix]
}
```

```
resource "azurerm_subnet" "bastion_subnet" {
  name                = "AzureBastionSubnet"
  resource_group_name = azurerm_resource_group.rg_hub.name
  virtual_network_name = azurerm_virtual_network.hub_vnet.name
  address_prefixes    = [var.bastion_subnet_prefix]
}
```

Spoke Virtual Network

```
resource "azurerm_virtual_network" "spoke_vnet" {
  name                = var.spoke_vnet_name
  location            = azurerm_resource_group.rg_spoke.location
  resource_group_name = azurerm_resource_group.rg_spoke.name
  address_space       = var.spoke_vnet_address_space
}
```

Spoke Subnets

```
resource "azurerm_subnet" "app_gateway_subnet" {
  name                = "AppGatewaySubnet"
  resource_group_name = azurerm_resource_group.rg_spoke.name
  virtual_network_name = azurerm_virtual_network.spoke_vnet.name
  address_prefixes    = [var.app_gateway_subnet_prefix]
}
```

```
resource "azurerm_subnet" "app_service_subnet" {
  name                = "AppServiceIntegrationSubnet"
  resource_group_name = azurerm_resource_group.rg_spoke.name
  virtual_network_name = azurerm_virtual_network.spoke_vnet.name
  address_prefixes    = [var.app_service_subnet_prefix]
}
```

```

delegation {
  name = "appservicesdelegation"
  service_delegation {
    name      = "Microsoft.Web/serverFarms"
    actions = ["Microsoft.Network/virtualNetworks/subnets/join/action"]
  }
}

resource "azurerm_subnet" "private_endpoints_subnet" {
  name                        = "PrivateEndpointSubnet"
  resource_group_name       = azurerm_resource_group.rg_spoke.name
  virtual_network_name      = azurerm_virtual_network.spoke_vnet.name
  address_prefixes          = [var.private_endpoints_subnet_prefix]
  private_endpoint_network_policies_enabled = false # Disable for simplicity, cons
}

```

Virtual Network Peering

```

resource "azurerm_virtual_network_peering" "hub_to_spoke" {
  name                        = "hub-to-spoke"
  resource_group_name       = azurerm_resource_group.rg_hub.name
  remote_virtual_network_id  = azurerm_virtual_network.spoke_vnet.id
  virtual_network_name      = azurerm_virtual_network.hub_vnet.name
  allow_virtual_network_access = true
  allow_forwarded_traffic    = true
  allow_gateway_transit      = false
}

resource "azurerm_virtual_network_peering" "spoke_to_hub" {
  name                        = "spoke-to-hub"
  resource_group_name       = azurerm_resource_group.rg_spoke.name
  remote_virtual_network_id  = azurerm_virtual_network.hub_vnet.id
  virtual_network_name      = azurerm_virtual_network.spoke_vnet.name
  allow_virtual_network_access = true
  allow_forwarded_traffic    = true
  use_remote_gateways        = false
}

```

Azure Firewall

```

resource "azurerm_public_ip" "firewall_pip" {
  name                = "fw-pip"
  location            = azurerm_resource_group.rg_hub.location
  resource_group_name = azurerm_resource_group.rg_hub.name
  allocation_method   = "Static"
  sku                 = "Standard"
}

resource "azurerm_firewall" "azure_firewall" {
  name = "azure-firewall"
}

```

```

location          = azurerm_resource_group.rg_hub.location
resource_group_name = azurerm_resource_group.rg_hub.name
sku_name          = "AZFW_VNet"
sku_tier          = "Standard"
ip_configuration {
  name              = "fw-config"
  subnet_id         = azurerm_subnet.firewall_subnet.id
  private_ip_address_allocation = "Dynamic"
  public_ip_address_id = azurerm_public_ip.firewall_pip.id
}
}

```

Azure Bastion

```

resource "azurerm_public_ip" "bastion_pip" {
  name              = "bastion-pip"
  location          = azurerm_resource_group.rg_hub.location
  resource_group_name = azurerm_resource_group.rg_hub.name
  allocation_method = "Static"
  sku               = "Standard"
}

```

```

resource "azurerm_bastion_host" "bastion" {
  name              = "azure-bastion"
  location          = azurerm_resource_group.rg_hub.location
  resource_group_name = azurerm_resource_group.rg_hub.name
  ip_configuration {
    name              = "bastion-config"
    subnet_id         = azurerm_subnet.bastion_subnet.id
    public_ip_address_id = azurerm_public_ip.bastion_pip.id
  }
}

```

Application Gateway with WAF

```

resource "azurerm_public_ip" "appgw_pip" {
  name              = "appgw-pip"
  location          = azurerm_resource_group.rg_spoke.location
  resource_group_name = azurerm_resource_group.rg_spoke.name
  allocation_method = "Static"
  sku               = "Standard"
}

```

```

resource "azurerm_application_gateway" "app_gateway" {
  name              = "app-gateway"
  location          = azurerm_resource_group.rg_spoke.location
  resource_group_name = azurerm_resource_group.rg_spoke.name
  sku {
    name      = "WAF_v2"
    tier       = "WAF_v2"
    capacity  = 2
  }
}

```

```

}
gateway_ip_configuration {
  name      = "appgateway-ip-config"
  subnet_id = azurerm_subnet.app_gateway_subnet.id
}
frontend_port {
  name = "http-port"
  port = 80
}
frontend_ip_configuration {
  name                = "appgateway-frontent-ip"
  public_ip_address_id = azurerm_public_ip.appgw_pip.id
}
backend_address_pool {
  name = "appservice-backendpool"
  # Backend addresses will be dynamically added later based on App Service insta
}
backend_http_settings {
  name                = "http-settings"
  port                = 80
  protocol            = "Http"
  cookie_based_affinity = "Disabled"
  request_timeout     = 30
}
http_listener {
  name                = "http-listener"
  frontend_ip_configuration_name = "appgateway-frontent-ip"
  frontend_port_name      = "http-port"
  protocol                = "Http"
}
request_routing_rule {
  name                = "http-routing-rule"
  priority            = 100
  http_listener_name  = "http-listener"
  backend_address_pool_name = "appservice-backendpool"
  backend_http_settings_name = "http-settings"
  rule_type           = "Basic"
}
}

```

App Service Plan

```

resource "azurerm_app_service_plan" "app_service_plan" {
  name                = "app-service-plan"
  location            = azurerm_resource_group.rg_spoke.location
  resource_group_name = azurerm_resource_group.rg_spoke.name
  sku {
    tier = "Isolated" # Consider Premium or other tiers
    size = "I1v2"
  }
}

```



```
    per_site_scaling = false
    is_xenon          = false
    reserved          = false
}
```

App Service Instances (Conceptual - Requires more detailed configuration for zone

```
resource "azurerm_app_service" "app_service_zone1" {
  name                        = "app-service-zone1"
  location                   = azurerm_resource_group.rg_spoke.location
  resource_group_name        = azurerm_resource_group.rg_spoke.name
  app_service_plan_id        = azurerm_app_service_plan.app_service_plan.id
  https_only                 = true
```

... other configurations like connection strings, etc.

```
  site_config {
    always_on = true
    # ... other site configurations
    ip_restrictions {
      action = "Allow"
      priority = 100
      service_tag = "AppService" # Allow inbound from App Gateway subnet
      virtual_network_subnet_id = azurerm_subnet.app_gateway_subnet.id
    }
  }
}
```

```
resource "azurerm_app_service" "app_service_zone2" {
  name                        = "app-service-zone2"
  location                   = azurerm_resource_group.rg_spoke.location
  resource_group_name        = azurerm_resource_group.rg_spoke.name
  app_service_plan_id        = azurerm_app_service_plan.app_service_plan.id
  https_only                 = true
```

... other configurations

```
  site_config {
    always_on = true
    # ... other site configurations
    ip_restrictions {
      action = "Allow"
      priority = 100
      service_tag = "AppService"
      virtual_network_subnet_id = azurerm_subnet.app_gateway_subnet.id
    }
  }
}
```

```
resource "azurerm_app_service" "app_service_zone3" {
  name                        = "app-service-zone3"
  location                   = azurerm_resource_group.rg_spoke.location
  resource_group_name        = azurerm_resource_group.rg_spoke.name
  app_service_plan_id        = azurerm_app_service_plan.app_service_plan.id
```

```

https_only          = true
# ... other configurations
site_config {
  always_on = true
  # ... other site configurations
  ip_restrictions {
    action = "Allow"
    priority = 100
    service_tag = "AppService"
    virtual_network_subnet_id = azurerm_subnet.app_gateway_subnet.id
  }
}
}

# Azure Storage Account (for client app deployment)
resource "azurerm_storage_account" "client_storage" {
  name                        = "clientstorageacc${random_id.storage_suffix.hex}"
  resource_group_name        = azurerm_resource_group.rg_spoke.name
  location                   = azurerm_resource_group.rg_spoke.location
  account_tier               = var.storage_account_tier
  account_replication_type   = var.storage_account_replication_type
}

# Azure Key Vault
resource "azurerm_key_vault" "key_vault" {
  name                        = "keyvault-${random_id.keyvault_suffix.hex}"
  location                   = azurerm_resource_group.rg_spoke.location
  resource_group_name        = azurerm_resource_group.rg_spoke.name
  tenant_id                  = data.azurerm_client_config.current.tenant_id
  sku_name                   = "standard"
  purge_protection_enabled   = false # Consider enabling for production
  soft_delete_retention_days = 7     # Consider adjusting for production

  access_policy {
    tenant_id = data.azurerm_client_config.current.tenant_id
    object_id = data.azurerm_client_config.current.object_id
    secret_permissions = [
      "Get", "List", "Set", "Delete", "Recover", "Backup", "Restore",
    ]
    key_permissions = [
      "Get", "Create", "Delete", "List", "Recover", "Backup", "Restore", "Sign", "
    ]
    certificate_permissions = [
      "Get", "List", "Delete", "Create", "Import", "Update", "ManageContacts", "Ge
    ]
  }
}
}

# Azure Container Registry

```

```
resource "azurerm_container_registry" "acr" {
  name                = "acr${random_id.acr_suffix.hex}"
  location             = azurerm_resource_group.rg_spoke.location
  resource_group_name = azurerm_resource_group.rg_spoke.name
  sku                 = "Standard"
  admin_enabled       = false # Consider enabling for specific scenarios
}
```

Azure AI Search

```
resource "azurerm_search_service" "ai_search" {
  name                = "aisearch-${random_id.aisearch_suffix.hex}"
  resource_group_name = azurerm_resource_group.rg_spoke.name
  location            = azurerm_resource_group.rg_spoke.location
  sku {
    name = "standard"
  }
}
```

Azure OpenAI Service

```
resource "azurerm_cognitive_account" "openai" {
  name                = "openai-${random_id.openai_suffix.hex}"
  location            = var.location # Check for Azure OpenAI availability i
  resource_group_name = azurerm_resource_group.rg_spoke.name
  kind               = "OpenAI"
  sku_name           = "S0" # Choose appropriate SKU
  custom_subdomain_name = "openai-${random_id.openai_suffix.hex}"
  public_network_access_enabled = false
}
```

Log Analytics Workspace

```
resource "azurerm_log_analytics_workspace" "log_analytics" {
  name                = "law-${random_id.loganalytics_suffix.hex}"
  location            = var.location
  resource_group_name = var.resource_group_name_spoke
  sku                 = "PerGB2018"
  retention_in_days   = 30
}
```

Random ID for Unique Names

```
resource "random_id" "storage_suffix" {
  byte_length = 4
}

resource "random_id" "keyvault_suffix" {
  byte_length = 4
}

resource "random_id" "acr_suffix" {
  byte_length = 4
}
```

```

}

resource "random_id" "aisearch_suffix" {
  byte_length = 4
}

resource "random_id" "openai_suffix" {
  byte_length = 4
}

resource "random_id" "loganalytics_suffix" {
  byte_length = 4
}

# Data Source for Current Client Configuration
data "azurerm_client_config" "current" {}

# Outputs (defined in outputs.tf)
output "hub_vnet_id" {
  value = azurerm_virtual_network.hub_vnet.id
}

output "spoke_vnet_id" {
  value = azurerm_virtual_network.spoke_vnet.id
}

output "app_gateway_public_ip" {
  value = azurerm_public_ip.appgw_pip.ip_address
}

output "key_vault_uri" {
  value = azurerm_key_vault.key_vault.vault_uri
}

output "acr_login_server" {
  value = azurerm_container_registry.acr.login_server
}

output "ai_search_endpoint" {
  value = azurerm_search_service.ai_search.primary_endpoint
}

output "openai_endpoint" {
  value = azurerm_cognitive_account.openai.endpoint
}

```

variables.tf:

```
variable "location" {
  description = "The Azure region to deploy resources into."
  type       = string
}

variable "resource_group_name_hub" {
  description = "Name of the resource group for the Hub network."
  type       = string
}

variable "resource_group_name_spoke" {
  description = "Name of the resource group for the Spoke network and workload res
  type       = string
}

variable "hub_vnet_name" {
  description = "Name of the Hub Virtual Network."
  type       = string
}

variable "hub_vnet_address_space" {
  description = "Address space for the Hub Virtual Network."
  type       = list(string)
}

variable "firewall_subnet_prefix" {
  description = "Address prefix for the Azure Firewall subnet."
  type       = string
}

variable "bastion_subnet_prefix" {
  description = "Address prefix for the Azure Bastion subnet."
  type       = string
}

variable "spoke_vnet_name" {
  description = "Name of the Spoke Virtual Network."
  type       = string
}

variable "spoke_vnet_address_space" {
  description = "Address space for the Spoke Virtual Network."
  type       = list(string)
}

variable "app_gateway_subnet_prefix" {
  description = "Address prefix for the Application Gateway subnet."
  type       = string
}
```

```

}

variable "app_service_subnet_prefix" {
  description = "Address prefix for the App Service integration subnet."
  type        = string
}

variable "private_endpoints_subnet_prefix" {
  description = "Address prefix for the Private Endpoints subnet."
  type        = string
}

variable "storage_account_tier" {
  description = "Tier for the Storage Account."
  type        = string
}

variable "storage_account_replication_type" {
  description = "Replication type for the Storage Account."
  type        = string
}

variable "app_service_plan_sku_name" {
  description = "SKU name for the App Service Plan."
  type        = string
}

variable "app_service_plan_capacity" {
  description = "Capacity for the App Service Plan."
  type        = number
}

```

outputs.tf:

```

output "hub_vnet_id" {
  description = "The ID of the Hub Virtual Network."
  value       = azurerm_virtual_network.hub_vnet.id
}

output "spoke_vnet_id" {
  description = "The ID of the Spoke Virtual Network."
  value       = azurerm_virtual_network.spoke_vnet.id
}

output "app_gateway_public_ip" {
  description = "The public IP address of the Application Gateway."
  value       = azurerm_public_ip.appgw_pip.ip_address
}

```

```

}

output "key_vault_uri" {
  description = "The URI of the Key Vault."
  value       = azurerm_key_vault.key_vault.vault_uri
}

output "acr_login_server" {
  description = "The login server of the Azure Container Registry."
  value       = azurerm_container_registry.acr.login_server
}

output "ai_search_endpoint" {
  description = "The primary endpoint of the Azure AI Search service."
  value       = azurerm_search_service.ai_search.primary_endpoint
}

output "openai_endpoint" {
  description = "The endpoint of the Azure OpenAI service."
  value       = azurerm_cognitive_account.openai.endpoint
}

```

Explanation and Next Steps:

1. **Resource Groups:** The code starts by creating resource groups to organize the Hub and Spoke network resources.
2. **Virtual Networks and Subnets:** It defines the Hub and Spoke VNets with their respective subnets, including dedicated subnets for Azure Firewall and Azure Bastion in the Hub, and for the Application Gateway and App Service in the Spoke.
3. **Virtual Network Peering:** Establishes peering between the Hub and Spoke VNets to enable communication.
4. **Azure Firewall:** Deploys an Azure Firewall instance in the Hub network for centralized security.
5. **Azure Bastion:** Creates an Azure Bastion host to securely connect to VMs within the virtual network.
6. **Application Gateway:** Sets up an Application Gateway with a basic HTTP listener and backend pool.
Note: The backend pool needs to be updated dynamically with the IP addresses of your App Service instances after they are created.
7. **App Service Plan and Instances:** Creates an App Service Plan and conceptual App Service instances.
Important:
 - **Availability Zones:** To deploy across availability zones, you need to ensure your App Service Plan and the underlying infrastructure support zones in your chosen region. You might need to configure instance counts or use deployment slots for zone awareness.
 - **Backend Pool Integration:** You'll need to add the App Service instances to the Application Gateway's backend pool, potentially using data sources or iterating through the App Service resources.

8. **Azure Storage, Key Vault, ACR, AI Search, OpenAI:** Provisions these PaaS services.
9. **Log Analytics Workspace:** Sets up a central location for logs.
10. **Random IDs:** Used to create unique names for certain resources.
11. **Outputs:** Displays important information like VNet IDs, public IPs, and service endpoints.

To Deploy:

1. **Save the code:** Save the code into three files: `main.tf`, `variables.tf`, and `outputs.tf`.
2. **Initialize Terraform:** Run `terraform init` in the directory where you saved the files.
3. **Plan the deployment:** Run `terraform plan` to see the changes that will be applied.
4. **Apply the configuration:** Run `terraform apply` and confirm with `yes` to create the resources in Azure.

Further Enhancements:

- **Managed Identity:** Implement managed identities for the App Service to securely access other Azure resources like Key Vault.
- **Private Endpoints:** Configure private endpoints for Azure Storage, Key Vault, ACR, AI Search, and OpenAI to enhance security and keep traffic within the private network. You'll need to create DNS records in your Private DNS Zones for these.
- **Network Security Groups (NSGs):** Add NSG rules to control traffic flow within the subnets.
- **Route Tables:** Implement user-defined routes (UDRs) to force traffic through the Azure Firewall.
- **Monitoring and Diagnostics:** Configure more detailed monitoring and diagnostics settings.
- **CI/CD Integration:** Integrate this Terraform code into a CI/CD pipeline for automated deployments.
- **Modules:** Break down the configuration into reusable modules for better organization and maintainability.
- **Variables:** Refine the variables to make the configuration more flexible.
- **Security Best Practices:** Implement more robust security measures based on your organization's policies.
- **Azure AI Foundry Specifics:** The "Azure AI Foundry" box is conceptual. You'll need to define the specific Azure services involved, such as Azure Machine Learning, Azure Kubernetes Service (AKS), or other compute and storage resources.

Remember to consult the official Azure documentation and Terraform provider documentation for more detailed information and configuration options for each resource.