

# 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



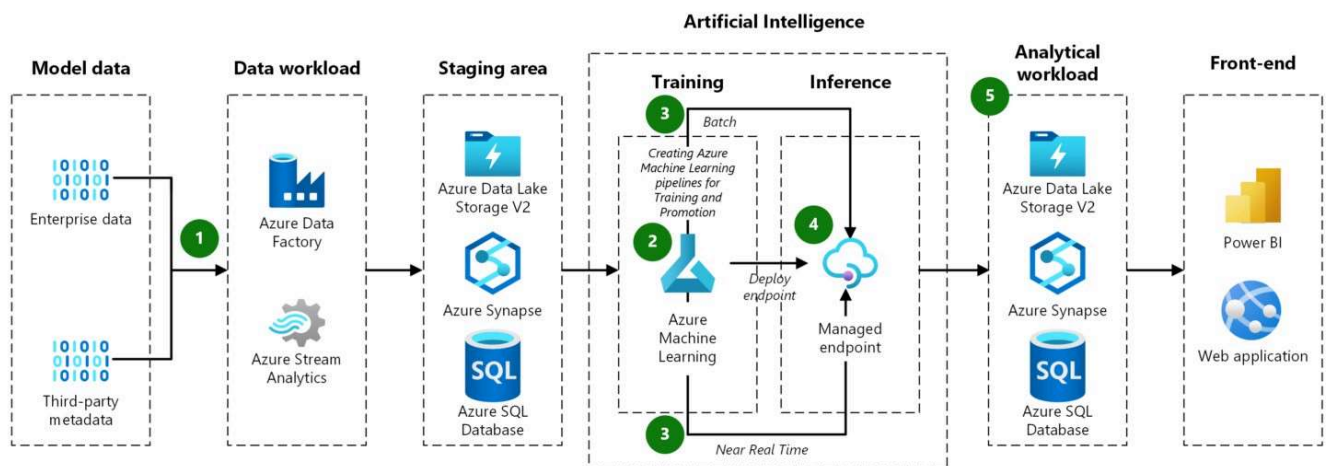
Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



many-models-machine-learning-azure.png 208.2KB



Uploaded Image

Enter your task/question for the AI Agent:

generate code for this

Analyze Image

## AI Response:

```
# Configure the Azure Provider
terraform {
  required_providers {
    azurerm = {
```

```

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

provider "azurerm" {
    features {}
}

# Replace with your desired resource group name and location
resource "azurerm_resource_group" "rg" {
    name      = "ai_pipeline_rg"
    location = "eastus"
}

# -----
# Data Workload
# -----

# Azure Data Factory
resource "azurerm_data_factory" "adf" {
    name                = "aيداتاپايفactory"
    location            = azurerm_resource_group.rg.location
    resource_group_name = azurerm_resource_group.rg.name
}

# Azure Stream Analytics Job (Basic example - you'll need to define inputs, output
resource "azurerm_stream_analytics_job" "asa" {
    name                = "aistreamanalyticsjob"
    resource_group_name = azurerm_resource_group.rg.name
    location            = azurerm_resource_group.rg.location
    streaming_units     = 1
    compatibility_level = "1.2"
    data_locale         = "en-US"
    events_late_arrival_max_delay_in_seconds = 5
    events_out_of_order_max_delay_in_seconds = 0
    events_out_of_order_policy = "Adjust"
    output_error_policy      = "Drop"
    sku_name                 = "Standard"
}

# -----
# Staging area
# -----

# Azure Data Lake Storage Gen2 Account
resource "azurerm_storage_account" "adls" {
    name = "aيداتالakestorageacc"

```

```

    resource_group_name      = azurerm_resource_group.rg.name
    location                 = azurerm_resource_group.rg.location
    account_tier             = "Standard"
    account_replication_type = "GRS"
    account_kind             = "StorageV2"
    is_hns_enabled          = true # Required for Data Lake Gen2
}

# Azure Synapse Workspace
resource "azurerm_synapse_workspace" "synapse" {
    name                        = "aisynapseworkspace"
    resource_group_name       = azurerm_resource_group.rg.name
    location                  = azurerm_resource_group.rg.location
    storage_data_lake_gen2_filesystem_id = azurerm_storage_account.adls.primary_blob
    sql_administrator_login    = "synapseadmin"
    sql_administrator_login_password = "YourSecurePassword!" # Replace with a strong
managed_virtual_network_enabled = true

    identity {
        type = "SystemAssigned"
    }
}

# Azure SQL Database
resource "azurerm_sql_server" "sqlserver_staging" {
    name                        = "aisqlserverstaging"
    resource_group_name       = azurerm_resource_group.rg.name
    location                  = azurerm_resource_group.rg.location
    version                   = "12.0" # Or latest
    administrator_login       = "sqladmin"
    administrator_login_password = "YourSecurePassword!" # Replace with a strong pas
    azuread_administrator {
        login_username         = "your_aad_user@yourdomain.com" # Replace with yo
        object_id              = "your_aad_user_object_id"      # Replace with your
        tenant_id              = "your_tenant_id"               # Replace with your
    }
}

resource "azurerm_sql_database" "sqlldb_staging" {
    name                        = "aisqldb_staging"
    resource_group_name       = azurerm_resource_group.rg.name
    location                  = azurerm_resource_group.rg.location
    server_id                 = azurerm_sql_server.sqlserver_staging.id
    requested_service_objective_name = "Basic" # Choose appropriate tier
}

# -----
# Training & Inference (Artificial Intelligence)
# -----

```

```

# Azure Machine Learning Workspace
resource "azurerm_machine_learning_workspace" "mlw" {
  name                        = "aimlworkspace"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  friendly_name              = "AI Machine Learning Workspace"
  storage_account_id         = azurerm_storage_account.adls.id
  key_vault_id               = azurerm_key_vault.kv.id
  application_insights_id    = azurerm_application_insights.appi.id
  identity {
    type = "SystemAssigned"
  }
}

# Azure Container Registry (often used with AML)
resource "azurerm_container_registry" "acr" {
  name                        = "aicontainerregistry"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  sku                        = "Basic"
  admin_enabled              = false
}

# Azure Key Vault (for secrets and keys)
resource "azurerm_key_vault" "kv" {
  name                        = "aikeyvault"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  tenant_id                  = "your_tenant_id" # Replace with your Azure 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 = "your_tenant_id" # Replace with your Azure Tenant ID
    object_id = "your_aad_user_object_id" # Replace with your Azure AD User Object ID
    secret_permissions = [
      "Get", "List",
    ]
  }
}

# Azure Application Insights (for monitoring)
resource "azurerm_application_insights" "appi" {
  name                        = "aiappinsights"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  application_type           = "web"

```

```

}

# -----
# Analytical workload
# -----

# Azure SQL Server for Analytical Workload
resource "azurerm_sql_server" "sqlserver_analytical" {
  name                        = "aisqlserveranalytical"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  version                    = "12.0" # Or latest
  administrator_login        = "sqladmin"
  administrator_login_password = "YourSecurePassword!" # Replace with a strong password
  azuread_administrator {
    login_username            = "your_aad_user@yourdomain.com" # Replace with your user
    object_id                 = "your_aad_user_object_id"      # Replace with your object ID
    tenant_id                 = "your_tenant_id"               # Replace with your tenant ID
  }
}

resource "azurerm_sql_database" "sqlldb_analytical" {
  name                        = "aisqldatabaseanalytical"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  server_id                  = azurerm_sql_server.sqlserver_analytical.id
  requested_service_objective_name = "Basic" # Choose appropriate tier
}

# -----
# Front-end (Illustrative - actual Power BI and Web App setup is more complex)
# -----

# Note: Power BI and Web App deployments are typically done through their respective
#       This section is just to acknowledge their presence in the diagram.

# Output the Synapse Workspace URL (for easy access)
output "synapse_workspace_url" {
  value = azurerm_synapse_workspace.synapse.connectivity_endpoints["Web"]
}

# Output the Machine Learning Workspace URL
output "ml_workspace_url" {
  value = azurerm_machine_learning_workspace.mlw.studio_url
}

```

## Explanation:

This Terraform code defines the infrastructure shown in the diagram. Here's a breakdown:

- **terraform block:** Configures the required Azure provider.
- **provider "azurerm" block:** Configures the Azure provider settings.
- **azurerm\_resource\_group :** Creates a resource group to contain all the resources.
- **Data Workload:**
  - **azurerm\_data\_factory :** Provisions an Azure Data Factory instance for data orchestration.
  - **azurerm\_stream\_analytics\_job :** Creates a basic Azure Stream Analytics job (you'll need to configure inputs, outputs, and the query).
- **Staging Area:**
  - **azurerm\_storage\_account :** Creates an Azure Data Lake Storage Gen2 account for storing data.
  - **azurerm\_synapse\_workspace :** Provisions an Azure Synapse Analytics workspace for data warehousing and big data analytics.
  - **azurerm\_sql\_server and azurerm\_sql\_database (staging):** Sets up an Azure SQL Database for staging data.
- **Training & Inference (Artificial Intelligence):**
  - **azurerm\_machine\_learning\_workspace :** Creates an Azure Machine Learning workspace for training and deploying models.
  - **azurerm\_container\_registry :** Provisions an Azure Container Registry to store Docker images (often used with Azure ML).
  - **azurerm\_key\_vault :** Creates an Azure Key Vault for securely storing secrets and keys.
  - **azurerm\_application\_insights :** Sets up Azure Application Insights for monitoring the AI pipeline.
- **Analytical Workload:**
  - **azurerm\_sql\_server and azurerm\_sql\_database (analytical):** Sets up an Azure SQL Database for analytical workloads.
- **Front-end:**
  - **Outputs:** Provides the URLs for the Synapse and Machine Learning workspaces for easy access.

### Important Notes:

- **Replace Placeholders:** Make sure to replace placeholders like `"YourSecurePassword!"` , `"your_aad_user@yourdomain.com"` , `"your_aad_user_object_id"` , and `"your_tenant_id"` with your actual values.
- **Security:** This is a basic infrastructure setup. For production environments, you'll need to implement more robust security measures, such as network security groups, private endpoints, and proper access control.
- **Configuration:** This code only creates the basic infrastructure. You'll need to further configure the services (e.g., create Data Factory pipelines, Synapse Spark pools, Machine Learning experiments, etc.).
- **Cost:** Be aware of the costs associated with the Azure services you are provisioning. Choose appropriate tiers and scaling options based on your needs.

- **Power BI and Web App:** The deployment and configuration of Power BI reports and web applications are typically done through their respective services and are not fully covered in this infrastructure code.

To use this code:

1. Make sure you have Terraform installed.
2. Save the code as a `.tf` file (e.g., `main.tf`).
3. Initialize Terraform: `terraform init`
4. Plan the deployment: `terraform plan`
5. Apply the configuration: `terraform apply`

This Terraform code provides a foundation for building the Azure AI pipeline depicted in the image. You will need to expand upon this base to create a fully functional and production-ready solution.