Below is a tight, do-now checklist plus a minimal **Terraform starter** you can run to stand up the core: hub VNet, Private DNS, Key Vault (CMK), ADLS Gen2 (HNS), and **Private Endpoints** for Storage & Key Vault. This gives you a secure “place to land” the data platform.

**Step 1 — Landing zone & security (Databricks-first)**

**1) Admin prep (one-time)**

* Create Entra ID groups (RBAC):
  + grp-aml-platform-ops, grp-aml-data-engineers, grp-aml-data-scientists, grp-aml-investigators.
* Create subscriptions (or pick existing): **platform**, **dev**, **prod**.
* Decide region (e.g., westeurope or switzerlandnorth) and CIDR (e.g., 10.10.0.0/16).

**2) Resource groups (platform subscription)**

* rg-aml-net-hub (network & private DNS)
* rg-aml-sec (Key Vault, identities)
* rg-aml-data (ADLS accounts)
* rg-aml-monitor (Log Analytics)

**3) Network & name resolution**

* **Hub VNet** vnet-aml-hub with subnets:
  + snet-pe (Private Endpoints)
  + (Reserve snet-dbx for Databricks VNet injection later)
* **Private DNS zones** & links:
  + privatelink.blob.core.windows.net
  + privatelink.dfs.core.windows.net
  + privatelink.vaultcore.azure.net

**4) Security foundations**

* **Key Vault** with soft-delete & purge protection.
* **Customer-Managed Key (CMK)** in Key Vault.
* **ADLS Gen2** storage account with HNS + **public network disabled**.
* **Private Endpoints** for **blob/dfs** and **vault**; map A records via Private DNS.
* **RBAC**:
  + grp-aml-platform-ops: **Contributor** on platform RGs.
  + grp-aml-data-engineers: **Storage Blob Data Contributor** on ADLS.
  + grp-aml-data-scientists: **Storage Blob Data Reader** (tighten later with ACLs).
  + grp-aml-investigators: **Storage Blob Data Reader** (gold only later).
* **Policy (optional but recommended)**:
  + Deny public network access on Storage/Key Vault.
  + Require Private Endpoints for Storage.
  + Allowed locations.

**Minimal Terraform starter (drop-in)**

**Files**

/infra

main.tf

variables.tf

outputs.tf

**main.tf**

terraform {

required\_version = ">= 1.6.0"

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "~> 3.116"

}

}

}

provider "azurerm" {

features {}

}

# ---------- Variables ----------

locals {

location = var.location

rg\_net\_name = "rg-aml-net-hub"

rg\_sec\_name = "rg-aml-sec"

rg\_data\_name = "rg-aml-data"

vnet\_name = "vnet-aml-hub"

pe\_subnet\_name = "snet-pe"

kv\_name = "kv-aml-plat"

sa\_name = var.sa\_name # must be globally unique & 3-24 chars

tags = { env = var.env, workload = "aml-platform", owner = var.owner }

}

# ---------- Resource Groups ----------

resource "azurerm\_resource\_group" "net" {

name = local.rg\_net\_name

location = local.location

tags = local.tags

}

resource "azurerm\_resource\_group" "sec" {

name = local.rg\_sec\_name

location = local.location

tags = local.tags

}

resource "azurerm\_resource\_group" "data" {

name = local.rg\_data\_name

location = local.location

tags = local.tags

}

# ---------- Networking: Hub VNet + Private Endpoint Subnet ----------

resource "azurerm\_virtual\_network" "hub" {

name = local.vnet\_name

location = local.location

resource\_group\_name = azurerm\_resource\_group.net.name

address\_space = [var.vnet\_cidr]

tags = local.tags

}

resource "azurerm\_subnet" "pe" {

name = local.pe\_subnet\_name

resource\_group\_name = azurerm\_resource\_group.net.name

virtual\_network\_name = azurerm\_virtual\_network.hub.name

address\_prefixes = [var.pe\_subnet\_cidr]

private\_endpoint\_network\_policies\_enabled = true

}

# ---------- Private DNS zones ----------

resource "azurerm\_private\_dns\_zone" "blob" {

name = "privatelink.blob.core.windows.net"

resource\_group\_name = azurerm\_resource\_group.net.name

}

resource "azurerm\_private\_dns\_zone" "dfs" {

name = "privatelink.dfs.core.windows.net"

resource\_group\_name = azurerm\_resource\_group.net.name

}

resource "azurerm\_private\_dns\_zone" "vault" {

name = "privatelink.vaultcore.azure.net"

resource\_group\_name = azurerm\_resource\_group.net.name

}

resource "azurerm\_private\_dns\_zone\_virtual\_network\_link" "blob\_link" {

name = "blob-link"

resource\_group\_name = azurerm\_resource\_group.net.name

private\_dns\_zone\_name = azurerm\_private\_dns\_zone.blob.name

virtual\_network\_id = azurerm\_virtual\_network.hub.id

registration\_enabled = false

}

resource "azurerm\_private\_dns\_zone\_virtual\_network\_link" "dfs\_link" {

name = "dfs-link"

resource\_group\_name = azurerm\_resource\_group.net.name

private\_dns\_zone\_name = azurerm\_private\_dns\_zone.dfs.name

virtual\_network\_id = azurerm\_virtual\_network.hub.id

registration\_enabled = false

}

resource "azurerm\_private\_dns\_zone\_virtual\_network\_link" "vault\_link" {

name = "vault-link"

resource\_group\_name = azurerm\_resource\_group.net.name

private\_dns\_zone\_name = azurerm\_private\_dns\_zone.vault.name

virtual\_network\_id = azurerm\_virtual\_network.hub.id

registration\_enabled = false

}

# ---------- Key Vault (for CMK) ----------

resource "azurerm\_key\_vault" "kv" {

name = local.kv\_name

location = local.location

resource\_group\_name = azurerm\_resource\_group.sec.name

tenant\_id = var.tenant\_id

sku\_name = "premium"

purge\_protection\_enabled = true

soft\_delete\_retention\_days = 90

enable\_rbac\_authorization = true

public\_network\_access\_enabled = false

tags = local.tags

}

# Example CMK key (RSA)

resource "azurerm\_key\_vault\_key" "cmk" {

name = "cmk-aml-storage"

key\_vault\_id = azurerm\_key\_vault.kv.id

key\_type = "RSA"

key\_size = 3072

key\_opts = ["decrypt","encrypt","sign","unwrapKey","verify","wrapKey"]

}

# ---------- ADLS Gen2 (HNS) with CMK ----------

resource "azurerm\_storage\_account" "adls" {

name = local.sa\_name

resource\_group\_name = azurerm\_resource\_group.data.name

location = local.location

account\_tier = "Premium"

account\_replication\_type = "ZRS"

account\_kind = "BlockBlobStorage" # or "StorageV2" (choose per your needs)

is\_hns\_enabled = true

min\_tls\_version = "TLS1\_2"

allow\_nested\_items\_to\_be\_public = false

public\_network\_access\_enabled = false

tags = local.tags

identity {

type = "SystemAssigned"

}

customer\_managed\_key {

key\_vault\_key\_id = azurerm\_key\_vault\_key.cmk.id

}

}

# ---------- Private Endpoints for Storage (blob + dfs) ----------

resource "azurerm\_private\_endpoint" "pe\_blob" {

name = "pe-${local.sa\_name}-blob"

location = local.location

resource\_group\_name = azurerm\_resource\_group.net.name

subnet\_id = azurerm\_subnet.pe.id

private\_service\_connection {

name = "blob-psc"

private\_connection\_resource\_id = azurerm\_storage\_account.adls.id

subresource\_names = ["blob"]

is\_manual\_connection = false

}

private\_dns\_zone\_group {

name = "blob-dnsgrp"

private\_dns\_zone\_ids = [azurerm\_private\_dns\_zone.blob.id]

}

}

resource "azurerm\_private\_endpoint" "pe\_dfs" {

name = "pe-${local.sa\_name}-dfs"

location = local.location

resource\_group\_name = azurerm\_resource\_group.net.name

subnet\_id = azurerm\_subnet.pe.id

private\_service\_connection {

name = "dfs-psc"

private\_connection\_resource\_id = azurerm\_storage\_account.adls.id

subresource\_names = ["dfs"]

is\_manual\_connection = false

}

private\_dns\_zone\_group {

name = "dfs-dnsgrp"

private\_dns\_zone\_ids = [azurerm\_private\_dns\_zone.dfs.id]

}

}

# ---------- Private Endpoint for Key Vault ----------

resource "azurerm\_private\_endpoint" "pe\_kv" {

name = "pe-${local.kv\_name}"

location = local.location

resource\_group\_name = azurerm\_resource\_group.net.name

subnet\_id = azurerm\_subnet.pe.id

private\_service\_connection {

name = "kv-psc"

private\_connection\_resource\_id = azurerm\_key\_vault.kv.id

subresource\_names = ["vault"]

is\_manual\_connection = false

}

private\_dns\_zone\_group {

name = "kv-dnsgrp"

private\_dns\_zone\_ids = [azurerm\_private\_dns\_zone.vault.id]

}

}

**variables.tf**

variable "tenant\_id" { type = string }

variable "location" { type = string default = "westeurope" }

variable "env" { type = string default = "plat" }

variable "owner" { type = string default = "aml-team" }

# Choose globally unique storage name (lowercase, 3-24 chars)

variable "sa\_name" { type = string }

# Networking

variable "vnet\_cidr" { type = string default = "10.10.0.0/16" }

variable "pe\_subnet\_cidr" { type = string default = "10.10.1.0/24" }

**outputs.tf**

output "key\_vault\_name" { value = azurerm\_key\_vault.kv.name }

output "storage\_account\_name" { value = azurerm\_storage\_account.adls.name }

output "vnet\_id" { value = azurerm\_virtual\_network.hub.id }

**How to run**

az login

az account set --subscription "<platform-subscription-id>"

cd infra

terraform init

terraform plan -var="tenant\_id=$(az account show --query tenantId -o tsv)" -var="sa\_name=staml<uniqueid>"

terraform apply -auto-approve -var="tenant\_id=$(az account show --query tenantId -o tsv)" -v

**What you’ll have after Step 1**

* A **locked-down landing zone**: ADLS Gen2 with **HNS + CMK**, **no public network**, reachable only via **Private Endpoints** in your hub VNet.
* **Key Vault** on private link for keys/secrets.
* **Private DNS** wired so PaaS resolves privately inside the hub.
* Clean **RBAC** scaffolding for engineers, scientists, and investigators.