Hello
my name is

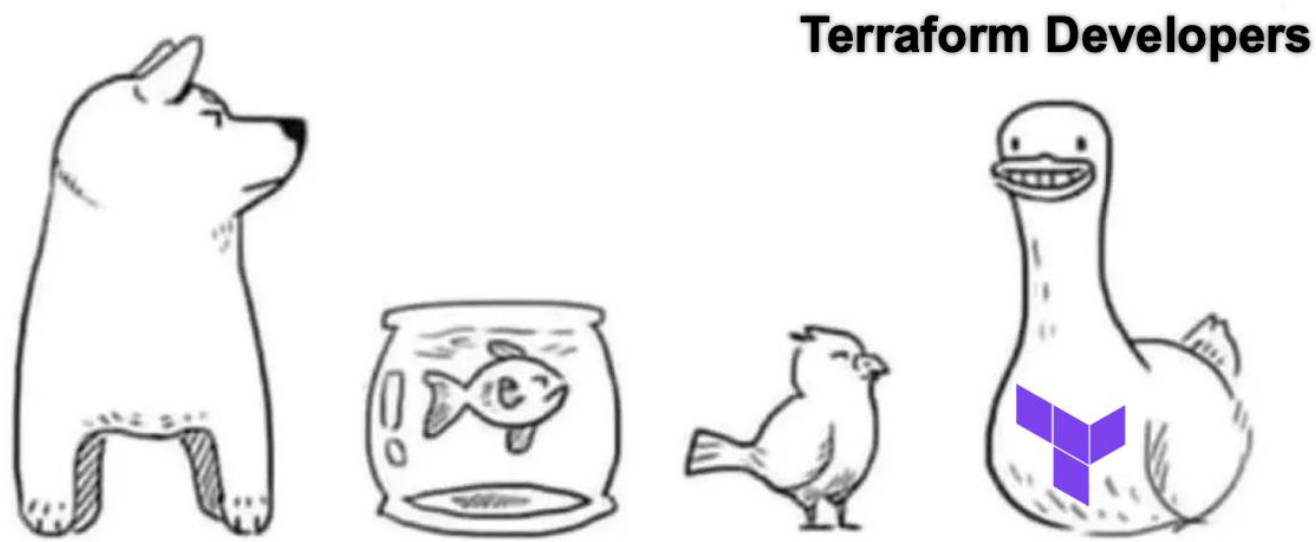
Mark Tinderholt

Principal Architect

Microsoft

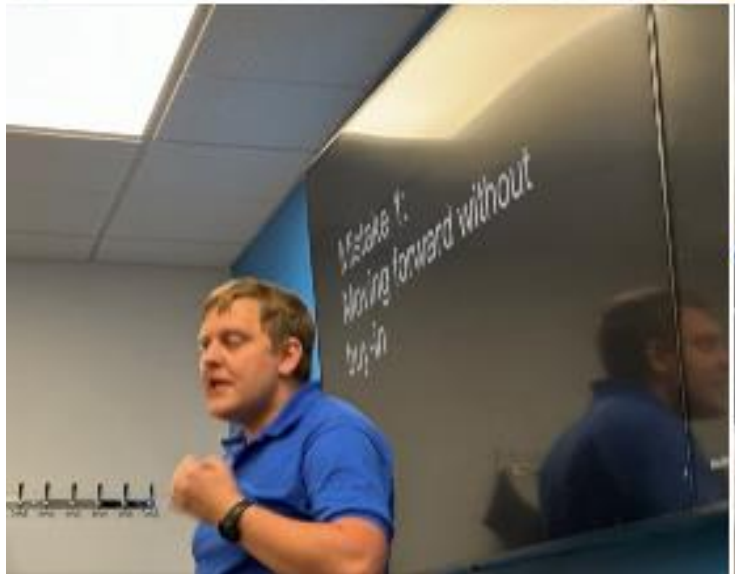
Microsoft Azure


I can't
I can't aws
I can't


Terraform Developers

cbus

cbus

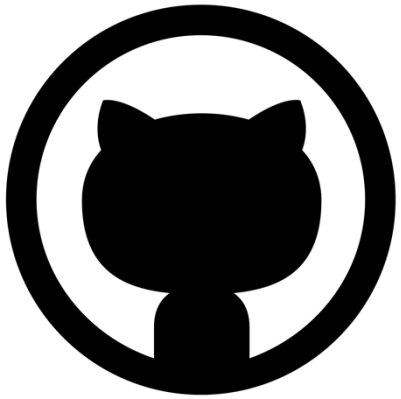


Azure Terraformer

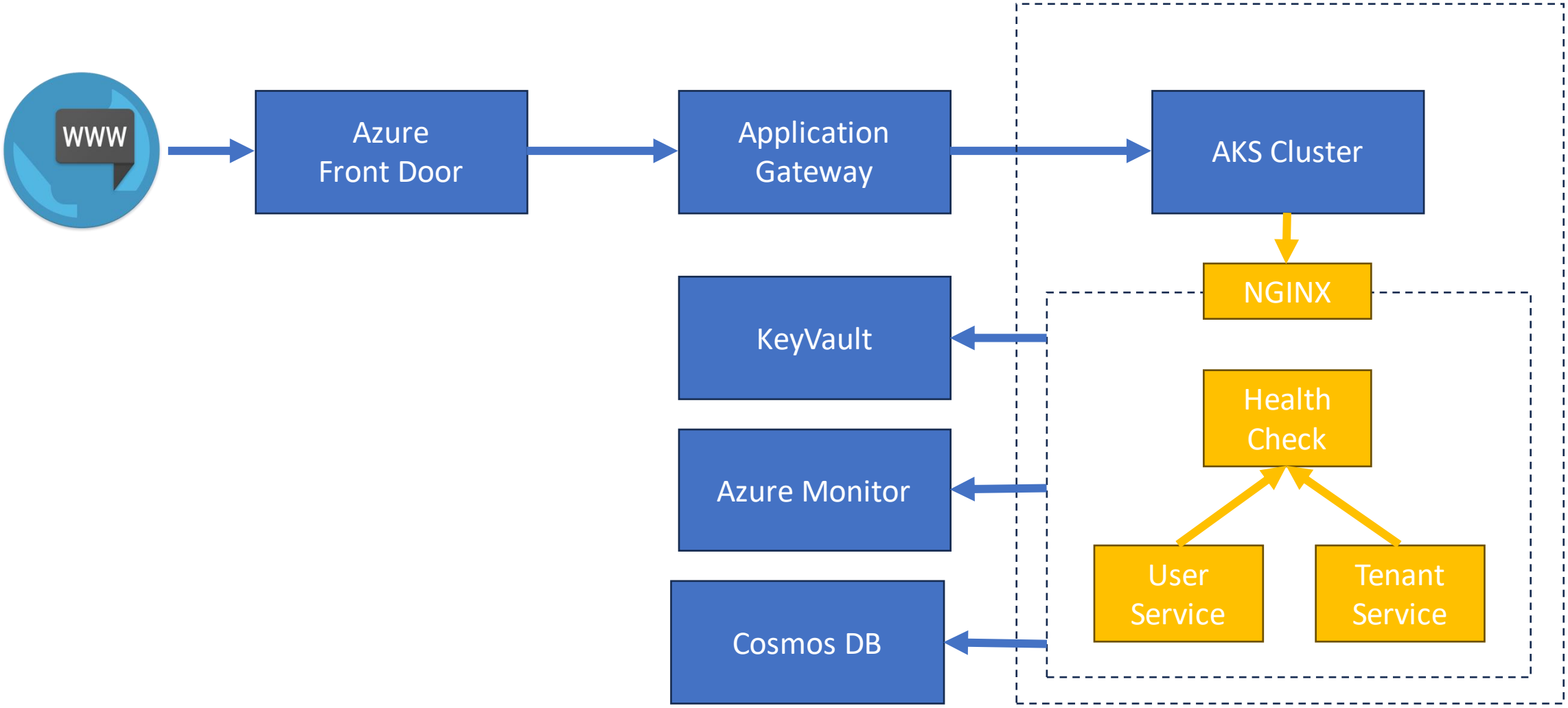
Terraforming AKS:

Design Considerations & Best Practices

Tech Stack


High Level Architecture



Codebase Structure

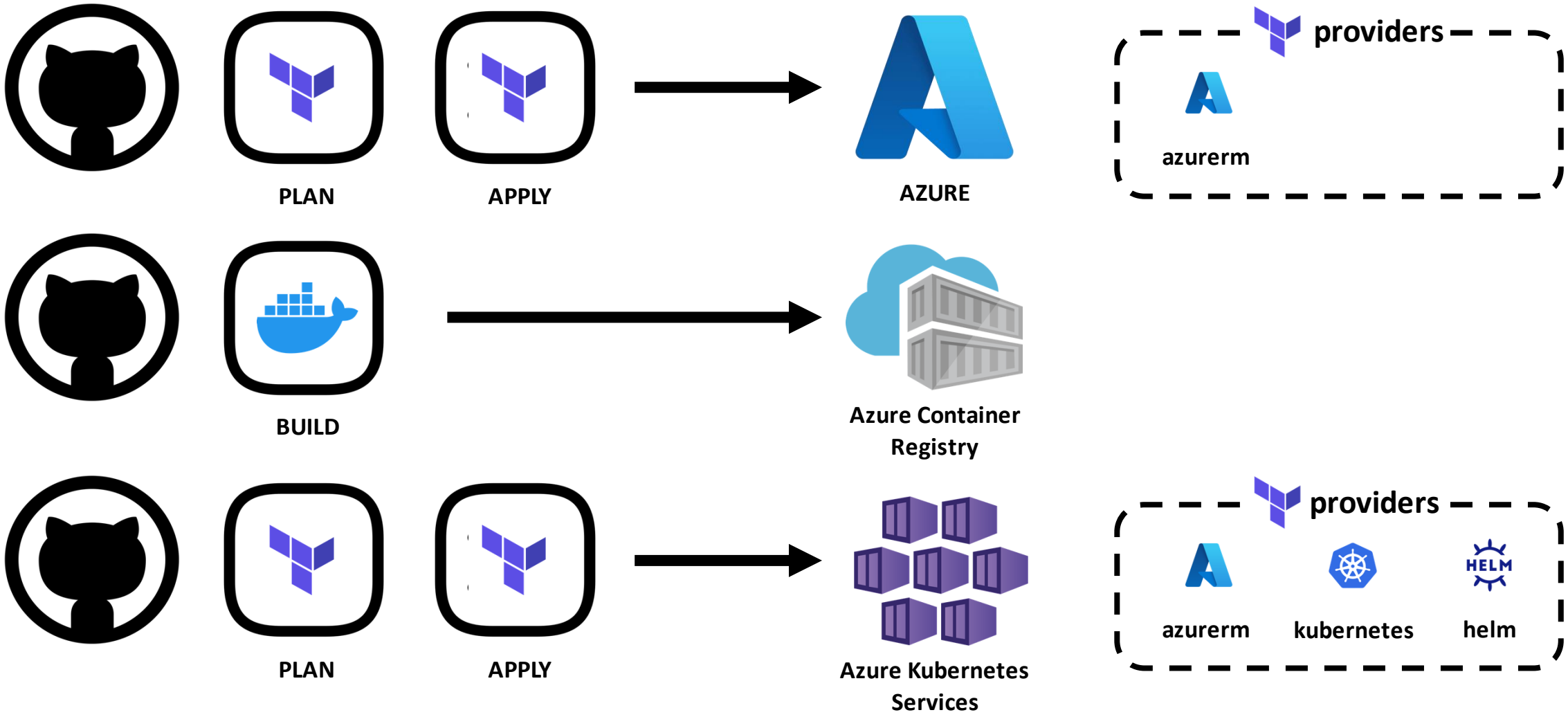
GitHub Actions
Workflows

Azure
Infrastructure

Kubernetes
Deployments

Application
Code

Pipeline Structure

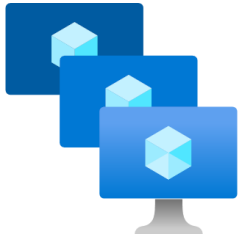


Agenda

1. Entra ID Integration

2. Private Networking

3. KeyVault Integration



4. Availability Zone Resiliency

5. Observability

6. Maintenance

Entra ID Integration

AKS Cluster


```
resource "azurerm_kubernetes_cluster" "main" {

  name                = "aks-${var.application_name}-${var.environment_name}-${random_string.main.result}"
  location            = azurerm_resource_group.main.location
  resource_group_name = azurerm_resource_group.main.name
```

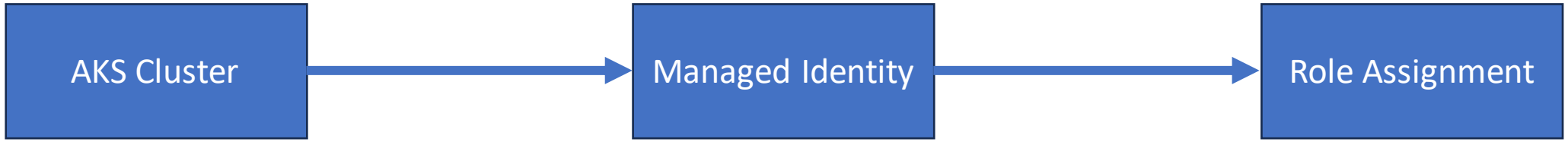
Managed Identity


```
resource "azurerm_user_assigned_identity" "aks_cluster" {
  location            = azurerm_resource_group.main.location
  resource_group_name = azurerm_resource_group.main.name
  name                = "mi-aks-${var.application_name}-${var.environment_name}-${random_string.main.result}"
}
```

Assign the Managed Identity

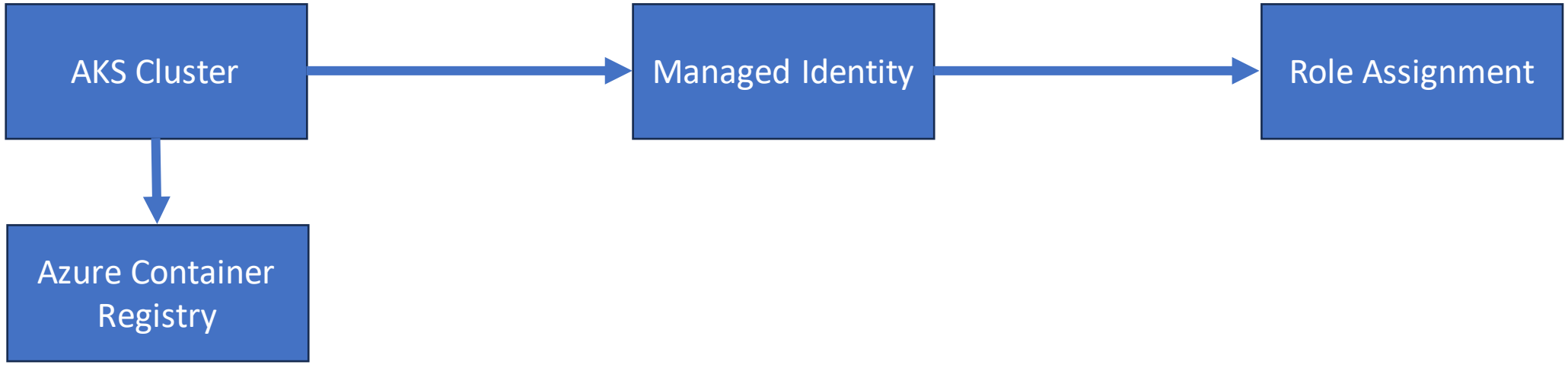

```
resource "azurerm_kubernetes_cluster" "main" {

  name                = "aks-${var.application_name}-${var.environment_name}-${random_string.main.result}"
  location            = azurerm_resource_group.main.location
  resource_group_name = azurerm_resource_group.main.name

  identity {
    type         = "UserAssigned"
    identity_ids = [azurerm_user_assigned_identity.aks_cluster.id]
  }
```

Create Cluster Role Assignment




```
resource "azurerm_role_assignment" "network_contributor" {
  scope                = azurerm_virtual_network.main.id
  role_definition_name = "Network Contributor"
  principal_id         = azurerm_user_assigned_identity.aks_cluster.principal_id
}
```

Connect to an Azure Container Registry




```
resource "azurerm_role_assignment" "acr_pull" {
  scope                = azurerm_container_registry.main.id
  role_definition_name = "AcrPull"
  principal_id         = azurerm_kubernetes_cluster.main.kubelet_identity[0].object_id
}
```

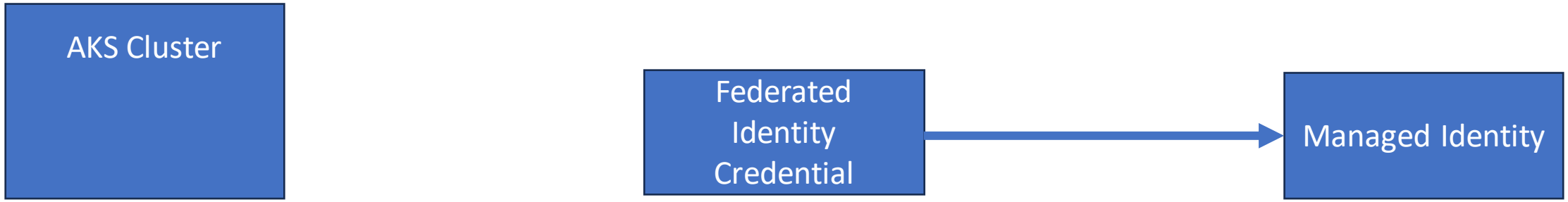
Connect to Entra ID


```
resource "azurerm_kubernetes_cluster" "main" {

  role_based_access_control_enabled = true

  azure_active_directory_role_based_access_control {
    managed                = true
    admin_group_object_ids = var.admin_groups
    azure_rbac_enabled     = true
  }
```

Enable Workload Identity

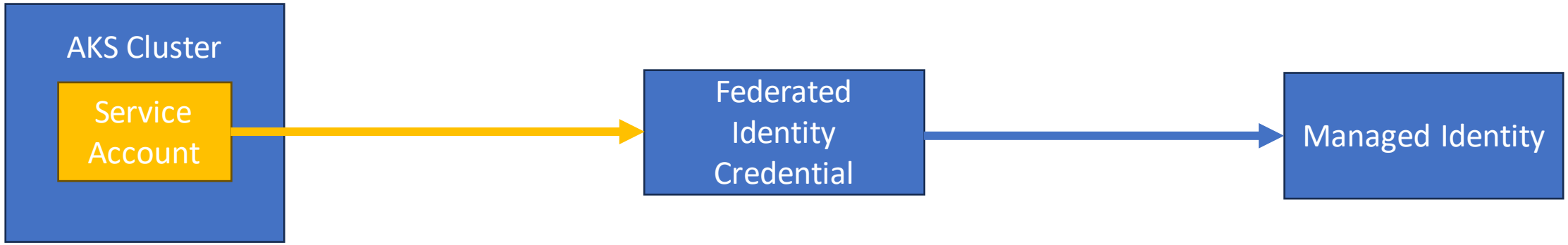

```
resource "azurerm_kubernetes_cluster" "main" {

  oidc_issuer_enabled             = true
  workload_identity_enabled       = true
```

Create Workload Identity



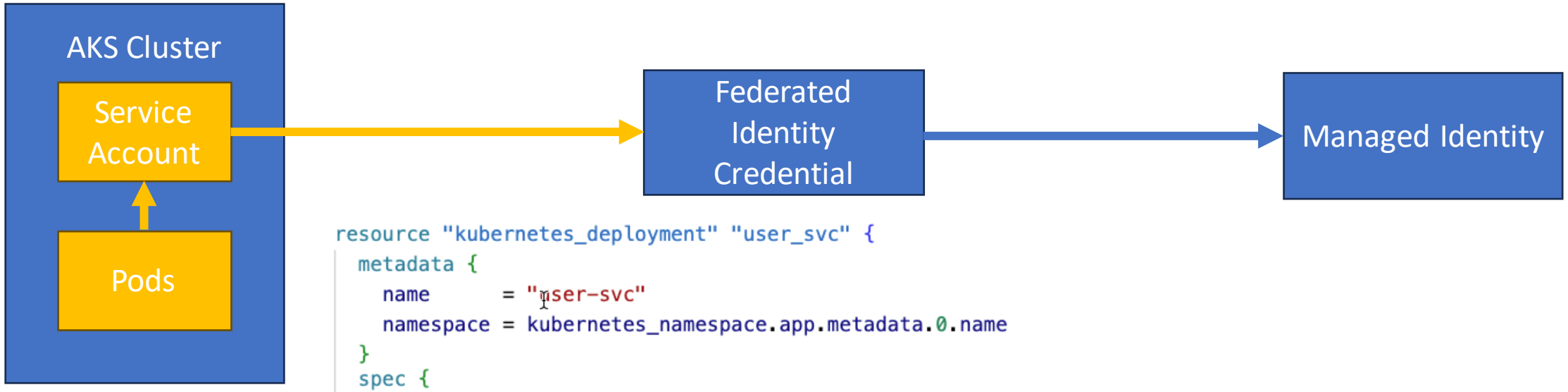

```
resource "azurerm_user_assigned_identity" "workload" {
  location            = azurerm_resource_group.main.location
  resource_group_name = azurerm_resource_group.main.name
  name                = "mi-workload-${var.application_name}-${var.environment_name}-${random_string.main.result}"
}

resource "azurerm_federated_identity_credential" "workload" {
  name                = azurerm_user_assigned_identity.workload.name
  resource_group_name = azurerm_resource_group.main.name
  audience            = ["api://AzureADTokenExchange"]
  issuer              = azurerm_kubernetes_cluster.main.oidc_issuer_url
  parent_id           = azurerm_user_assigned_identity.workload.id
  subject             = "system:serviceaccount:${var.k8s_namespace}:${var.k8s_service_account_name}"
}
```

Setup Kubernetes Service Account




```
resource "kubernetes_service_account" "workload" {
  metadata {
    name = var.k8s_service_account_name
    namespace = var.k8s_namespace
    annotations = {
      "azure.workload.identity/client-id" = var.workload_managed_identity_id
    }
  }
}
```

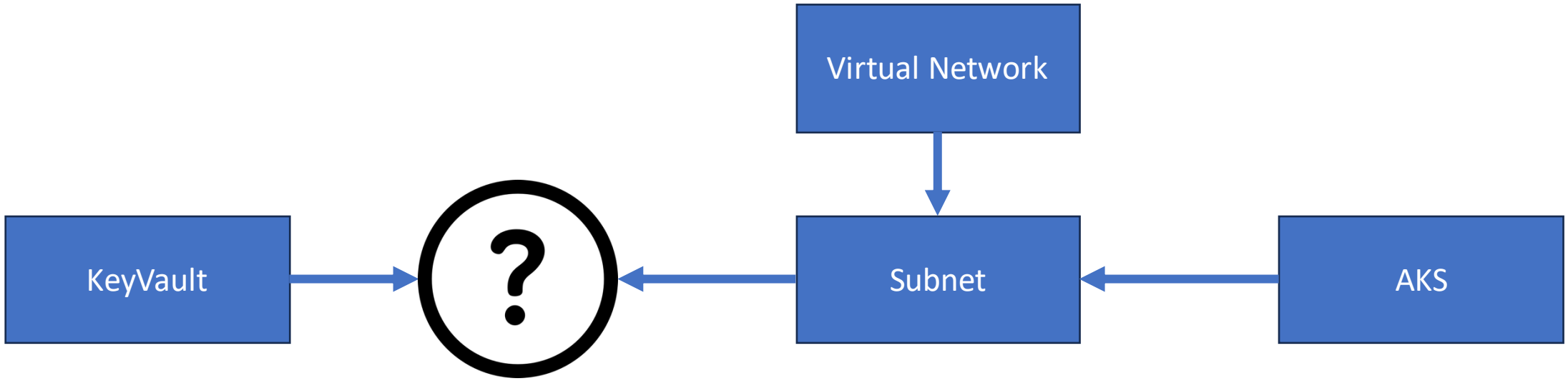
Connect Pods to the Service Account



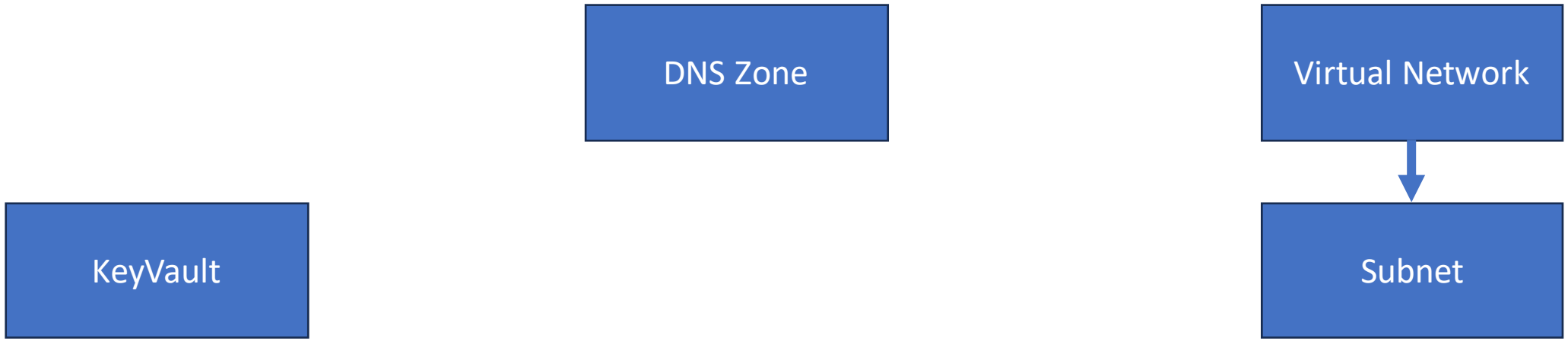

```
resource "kubernetes_deployment" "user_svc" {
  metadata {
    name      = "user-svc"
    namespace = kubernetes_namespace.app.metadata.0.name
  }
  spec {
    template {
      metadata {
        labels = {
          app                          = "user-svc"
          "azure.workload.identity/use" = "true"
        }
      }

      spec {
        service_account_name = kubernetes_service_account.workload.metadata.0.name
```

Private Networking

Where to begin?

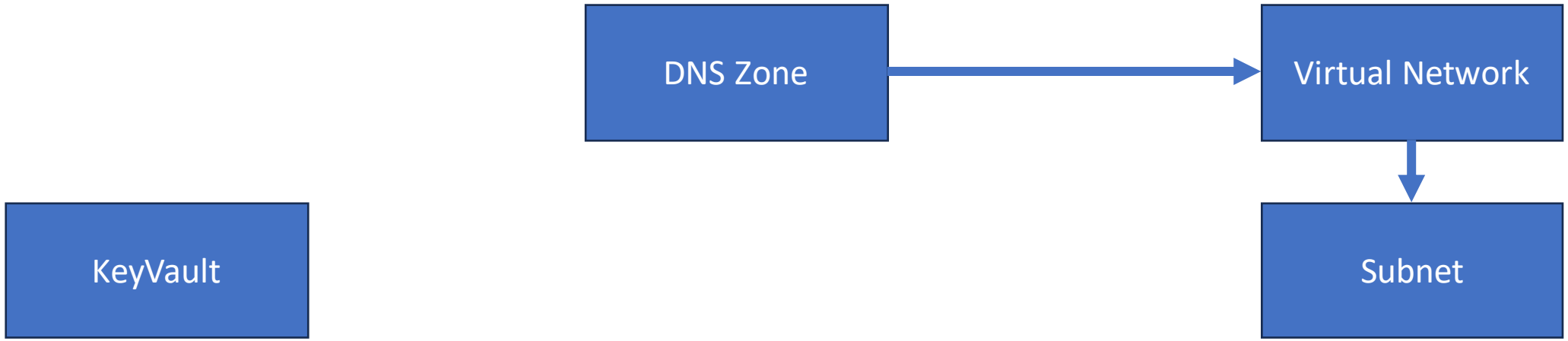


Create a DNS Zone



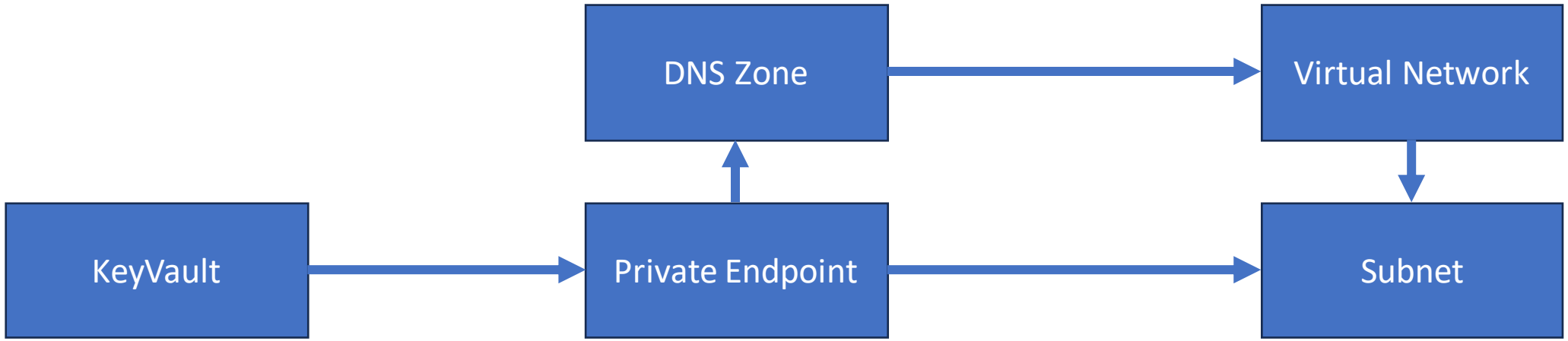

```
resource "azurerm_private_dns_zone" "keyvault" {
  name                = "privatelink.vaultcore.azure.net"
  resource_group_name = azurerm_resource_group.main.name
}
```

Link the DNS Zone to the Virtual Network




```
resource "azurerm_private_dns_zone_virtual_network_link" "keyvault_workload" {
  name                  = "dns-link-keyvault-workload"
  resource_group_name   = azurerm_resource_group.main.name
  private_dns_zone_name = azurerm_private_dns_zone.keyvault.name
  virtual_network_id    = azurerm_virtual_network.main.id
}
```

Create a Private Endpoint




```
resource "azurerm_private_endpoint" "keyvault" {
  subnet_id           = azurerm_subnet.shared.id

  private_service_connection {
    name                           = "${azurerm_key_vault.main.name}-link"
    private_connection_resource_id = azurerm_key_vault.main.id
    subresource_names              = ["vault"]
    is_manual_connection           = false
  }
  private_dns_zone_group {
    name                 = "vault-dns"
    private_dns_zone_ids = [azurerm_private_dns_zone.keyvault.id]
  }
```

KeyVault Integration

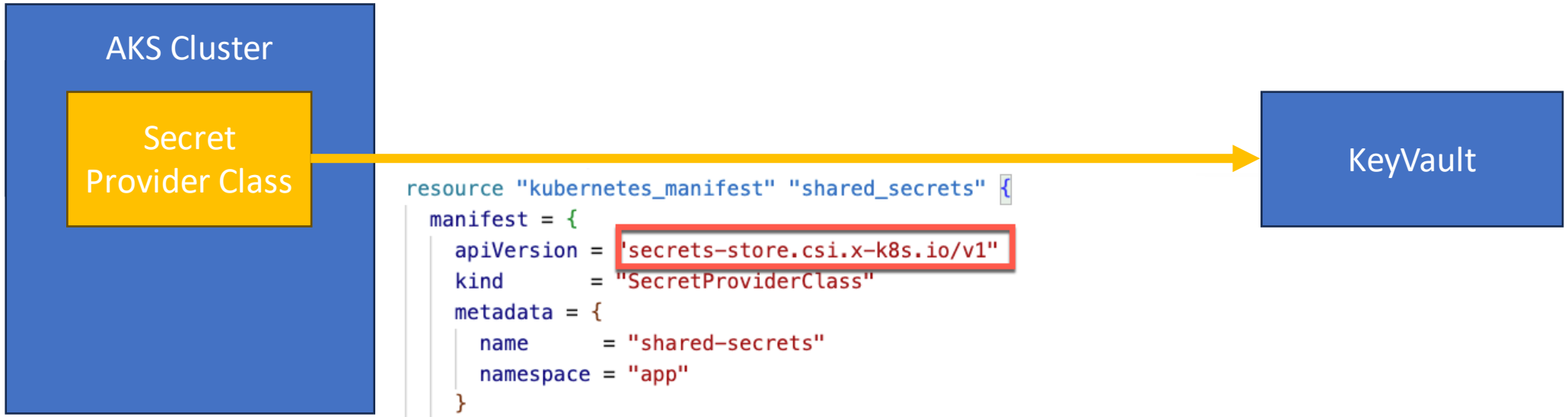
Enable the KeyVault Secrets Provider

AKS Cluster

KeyVault

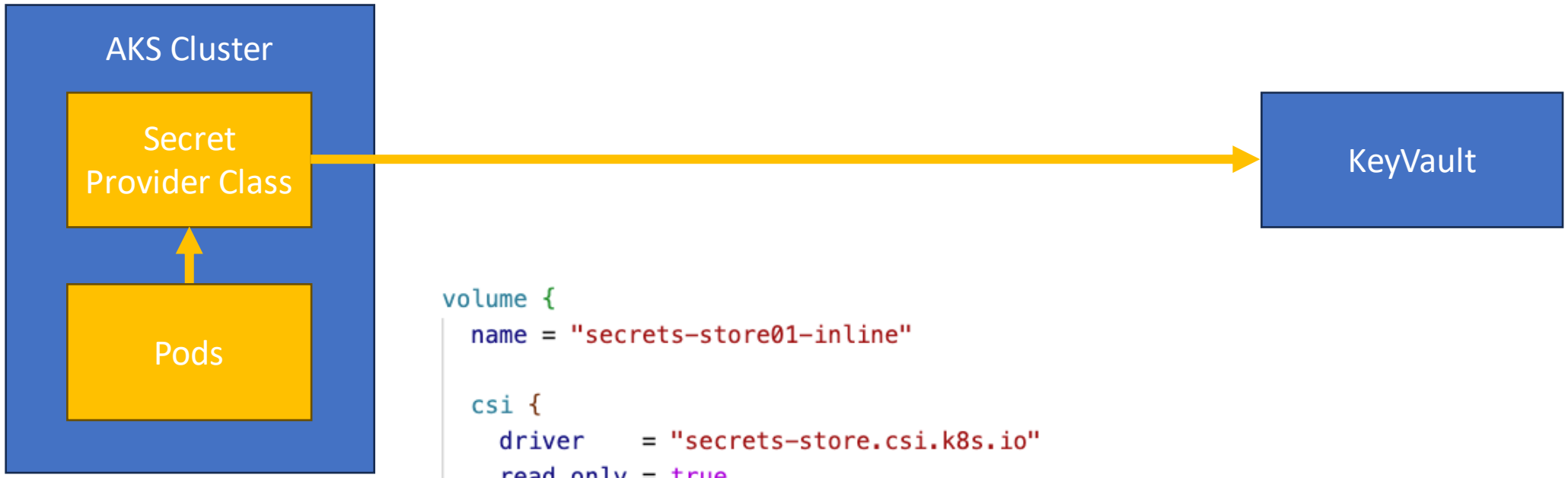
```
resource "azurerm_kubernetes_cluster" "main" {

  key_vault_secrets_provider {
    secret_rotation_enabled  = false
    secret_rotation_interval = "2m"
  }
```

Provision a Secret Provider Class

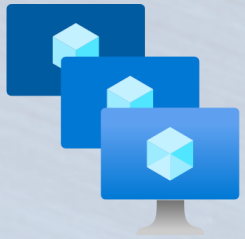



```
resource "kubernetes_manifest" "shared_secrets" {
  manifest = {
    apiVersion = 'secrets-store.csi.x-k8s.io/v1"
    kind       = "SecretProviderClass"
    metadata = {
      name      = "shared-secrets"
      namespace = "app"
    }
    spec = {
      provider = "azure"
      parameters = {
        usePodIdentity = "false"
        clientID       = var.workload_managed_identity_id
        keyvaultName   = var.keyvault_name
        cloudName      = ""
        objects        = <<OBJECTS
      array:
        - |
          objectName: app-insights-connection-string
          objectType: secret
```

Reference from Pods




```
volume {
  name = "secrets-store01-inline"

  csi {
    driver    = "secrets-store.csi.k8s.io"
    read_only = true

    volume_attributes = {
      secretProviderClass = kubernetes_manifest.shared_secrets.manifest.metadata.name
    }
  }
}
```



Availability Zone Resilliency

Provision Azure Resources with AZ Resiliency!

```
resource "azurerm_kubernetes_cluster_node_pool" "workload" {

  name                  = "npworkload"
  kubernetes_cluster_id = azurerm_kubernetes_cluster.main.id
  vm_size               = var.aks_configuration.workload_pool.sku
  enable_auto_scaling   = true
  os_sku                = "Mariner"
  vnet_subnet_id        = azurerm_subnet.workload.id
  zones                 = [1, 2, 3]
  node_count            = var.aks_configuration.workload_pool.capacity.ready
  min_count             = var.aks_configuration.workload_pool.capacity.min
  max_count             = var.aks_configuration.workload_pool.capacity.max

}
```

Zone-Isolated Node Pools

```
resource "azurerm_kubernetes_cluster_node_pool" "zone1" {

  name                  = "npzone1"
  kubernetes_cluster_id = azurerm_kubernetes_cluster.main.id
  vm_size               = var.aks_configuration.workload_pool.sku
  enable_auto_scaling   = true
  os_sku                = "Mariner"
  vnet_subnet_id        = azurerm_subnet.workload.id
  zones                 = [1]
  node_count            = var.aks_configurati
  min_count             = var.aks_configurati
  max_count             = var.aks_configurati

}
```

```
resource "azurerm_kubernetes_cluster_node_pool" "zone2" {

  name                  = "npzone1"
  kubernetes_cluster_id = azurerm_kubernetes_cluster.main.id
  vm_size               = var.aks_configuration.workload_pool.sku
  enable_auto_scaling   = true
  os_sku                = "Mariner"
  vnet_subnet_id        = azurerm_subnet.wor
  zones                 = [2]
  node_count            = var.aks_configurat
  min_count             = var.aks_configurat
  max_count             = var.aks_configurat

}
```

```
resource "azurerm_kubernetes_cluster_node_pool" "zone3" {

  name                  = "npzone1"
  kubernetes_cluster_id = azurerm_kubernetes_cluster.main.id
  vm_size               = var.aks_configuration.workload_pool.sku
  enable_auto_scaling   = true
  os_sku                = "Mariner"
  vnet_subnet_id        = azurerm_subnet.workload.id
  zones                 = [3]
  node_count            = var.aks_configuration.workload_pool.capacity.ready
  min_count             = var.aks_configuration.workload_pool.capacity.min
  max_count             = var.aks_configuration.workload_pool.capacity.max

}
```

AZ Resiliency Azure Policy


```
data "azurerm_policy_set_definition" "az_resiliency" {
  name = "130fb88f-0fc9-4678-bfe1-31022d71c7d5"
}

resource "azurerm_resource_group_policy_assignment" "az_resiliency" {
  name                 = "pol-az-resiliency"
  resource_group_id    = azurerm_resource_group.main.id
  policy_definition_id = data.azurerm_policy_set_definition.az_resiliency.id

  parameters = jsonencode({
    effect = {
      value = "Audit"
    }
    allow = {
      value = "Both"
    }
  })

}
```

AZ Resiliency Azure Policy

View assignment | Create remediation task | Create exemption | Activity Logs

Essentials

Name : --
Description : --
Assignment ID : /subscriptions/a8dc551f-cbe8-47e9-87c1-d9570ac6d69d/resourcegroups/rg-aztflab-dev-agyhy3jv/providers/micr...

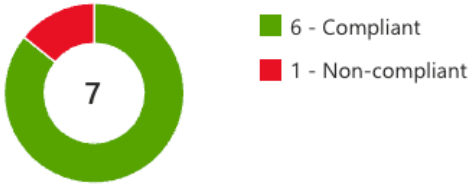
Scope : markti lab/rg-aztflab-dev-agyhy3jv
Definition : [Preview]: Resources should be Zone Resilient

Scope : **markti lab/rg-aztflab-dev-agyhy3jv**

Compliance state
Non-compliant

Overall resource compliance
86%
6 out of 7

Resources by compliance state



Non-compliant policies
1
out of 22

Policies | Non-compliant resources

Filter by policy name or definition ID...

Compliance state : **All compliance states**

Name	Effect Type	Compliance state	Non-Compliant Resources
[Preview]: Cosmos Database Accounts should be Zone Redundant	Audit	Non-compliant	1
[Preview]: Azure Kubernetes Service Managed Clusters should be Zone Rec	Audit	Compliant	0
[Preview]: Container Registry should be Zone Redundant	Audit	Compliant	0

Observability

Observability


```
resource "azurerm_kubernetes_cluster" "main" {

  microsoft_defender {
    log_analytics_workspace_id = azurerm_log_analytics_workspace.main.id
  }

  oms_agent {
    log_analytics_workspace_id = azurerm_log_analytics_workspace.main.id
  }
```

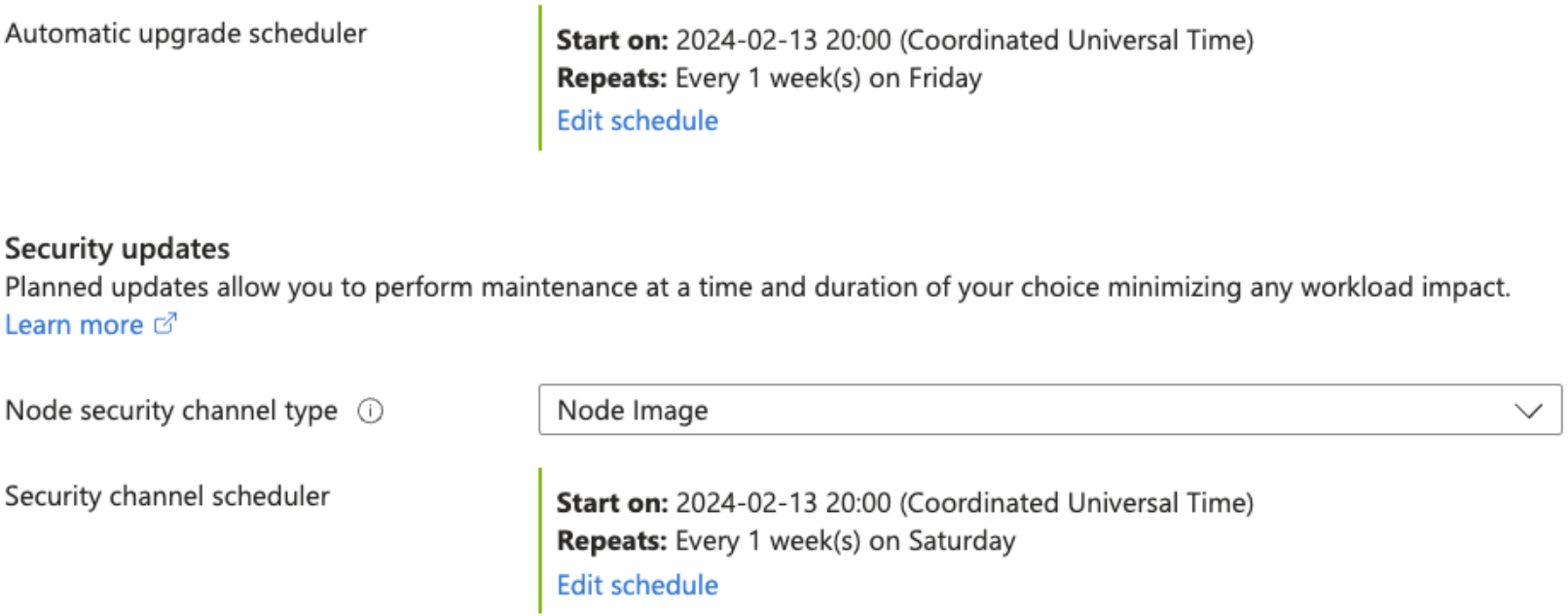
Azure Monitor Diagnostics

```
module "aks_monitor_diagnostic" {
  source  = "markti/azure-terraformer/azurerm//modules/monitor/diagnostic-setting/rando"
  version = "1.0.10"

  resource_id                = azurerm_kubernetes_cluster.main.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.main.id
  logs = [
    "kube-apiserver",
    "kube-audit",
    "kube-audit-admin",
    "kube-controller-manager",
    "kube-scheduler",
    "cluster-autoscaler",
    "cloud-controller-manager",
    "guard",
    "csi-azuredisk-controller",
    "csi-azurefile-controller",
    "csi-snapshot-controller"
  ]

}
```

Maintenance

Maintenance Windows

```hcl
resource "azurerm_kubernetes_cluster" "main" {
  automatic_channel_upgrade         = "patch"
  node_os_channel_upgrade           = "NodeImage"

  maintenance_window_auto_upgrade {
    frequency   = "Weekly"
    interval    = 1
    duration    = 4
    day_of_week = "Friday"
    utc_offset  = "-05:00"
    start_time  = "20:00"
  }

  maintenance_window_node_os {
    frequency   = "Weekly"
    interval    = 1
    duration    = 4
    day_of_week = "Saturday"
    utc_offset  = "-05:00"
    start_time  = "20:00"
  }
```




THANK
YOU