Azure Key Vault Overview

Azure Key Vault is a cloud service that securely stores and manages secrets, encryption keys, and certificates used by applications.

1. Key Vault Components

Secrets (🔑)

- Stores passwords, API keys, connection strings, and tokens.
- **Example:** A database password stored as a secret.

Keys (🔐)

- Stores encryption keys (RSA, EC) for data encryption and decryption.
- Used for disk encryption, SSL/TLS, and signing JWTs.

Certificates (📜)

- Manages SSL/TLS certificates from public CAs like DigiCert.
- Automates certificate renewal.

2. Key Vault Access Control

Access Methods

1. **RBAC (Role-Based Access Control)** → Manages permissions at the Azure level.
2. **Access Policies** → Directly control specific Key Vault actions (deprecated in newer versions).

Common RBAC Roles

- **Key Vault Administrator** → Full access to manage secrets, keys, and certificates.
- **Key Vault Secrets User** → Read secrets only.
- **Key Vault Crypto User** → Use keys but cannot see them.

3. Authentication Methods

Managed Identity (✅ Recommended)

- Azure services like VMs, Functions, and App Services can authenticate without storing credentials.

Service Principal (🔹 App Registration)

- Uses Client ID and Secret or Certificate authentication.

User Authentication (❌ Not Recommended)

- Requires manual login using az login.

4. Terraform Integration with Azure Key Vault

Storing and Retrieving Secrets in Terraform

```
resource "azurerm_key_vault" "example" {
  name                = "my-keyvault"
  resource_group_name = "my-rg"
  location            = "West Europe"
  sku_name            = "standard"
}

resource "azurerm_key_vault_secret" "db_password" {
  name         = "db-password"
  value        = "MySecurePassword123"
  key_vault_id = azurerm_key_vault.example.id
}

output "db_password" {
  value     = azurerm_key_vault_secret.db_password.value
  sensitive = true  # Hides output
}
```

✅ Ensures secrets are securely stored instead of hardcoding them in Terraform.

5. Retrieving Secrets in Azure DevOps Pipelines

Use Azure Key Vault Task to retrieve secrets:

```
- task: AzureKeyVault@2
  inputs:
    azureSubscription: 'MyServiceConnection'
    KeyVaultName: 'my-keyvault'
    SecretsFilter: '*'
```

✅ This allows secure access to secrets in CI/CD pipelines.

6. Best Practices

✅ Use Managed Identities instead of service principals. ✅ Enable soft delete to recover accidentally deleted secrets. ✅ Enable logging in Azure Monitor for auditing. ✅ Use RBAC roles instead of access policies for fine-grained control. ✅ Do not store secrets in Terraform state files—use Key Vault instead.