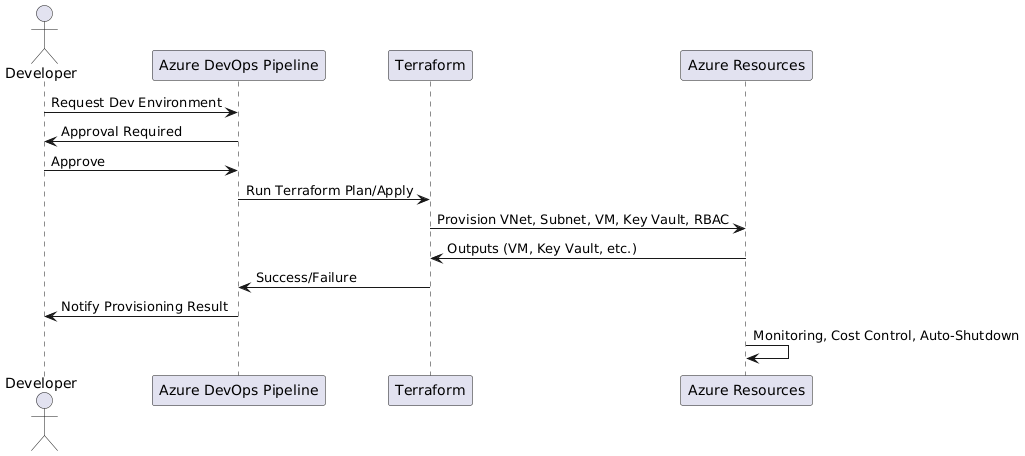
**# Step-by-Step Guide: Azure Self-Service Dev Environment Provisioning**

**This guide provides a detailed, step-by-step process with exact configuration for building a secure, automated self-service developer environment provisioning solution on Azure using Terraform and Azure DevOps.**

**Architecture Diagram**



**1. Prerequisites**

**- Azure Subscription with Owner/Contributor rights**

**- Azure DevOps Project**

**- Service Principal with sufficient permissions (for Terraform and pipeline)**

**- Azure Storage Account for Terraform state**

**- Registered VM image (e.g., Windows 10 or Ubuntu)**

**2. Create Terraform Modules**

**Directory Structure**

**infra/**

**main.tf**

**variables.tf**

**outputs.tf**

**modules/**

**dev\_env/**

**main.tf**

**variables.tf**

**outputs.tf**

**---------------------------------------------------------------------------------------------**

**infra/modules/dev\_env/variables.tf**

**---------------------------------------------------------------------------------------------**

**variable "resource\_group\_name" {}**

**variable "location" {}**

**variable "admin\_username" {}**

**variable "admin\_password" {}**

**variable "tenant\_id" {}**

**variable "dev\_user\_object\_id" {}**

**variable "tags" { type = map(string) }**

**--------------------------------------------------------------------------------------------**

**infra/modules/dev\_env/main.tf**

**--------------------------------------------------------------------------------------------**

**resource "azurerm\_resource\_group" "dev" {**

**name     = var.resource\_group\_name**

**location = var.location**

**tags     = var.tags**

**}**

**resource "azurerm\_virtual\_network" "dev" {**

**name                = "dev-vnet"**

**address\_space       = ["10.10.0.0/16"]**

**location            = azurerm\_resource\_group.dev.location**

**resource\_group\_name = azurerm\_resource\_group.dev.name**

**}**

**resource "azurerm\_subnet" "dev" {**

**name                 = "dev-subnet"**

**resource\_group\_name  = azurerm\_resource\_group.dev.name**

**virtual\_network\_name = azurerm\_virtual\_network.dev.name**

**address\_prefixes     = ["10.10.1.0/24"]**

**}**

**resource "azurerm\_network\_interface" "dev" {**

**name                = "dev-nic"**

**location            = azurerm\_resource\_group.dev.location**

**resource\_group\_name = azurerm\_resource\_group.dev.name**

**ip\_configuration {**

**name                          = "internal"**

**subnet\_id                     = azurerm\_subnet.dev.id**

**private\_ip\_address\_allocation = "Dynamic"**

**}**

**}**

**resource "azurerm\_windows\_virtual\_machine" "dev" {**

**name                = "dev-vm"**

**resource\_group\_name = azurerm\_resource\_group.dev.name**

**location            = azurerm\_resource\_group.dev.location**

**size                = "Standard\_B2ms"**

**admin\_username      = var.admin\_username**

**admin\_password      = var.admin\_password**

**network\_interface\_ids = [azurerm\_network\_interface.dev.id]**

**os\_disk {**

**caching              = "ReadWrite"**

**storage\_account\_type = "Standard\_LRS"**

**}**

**source\_image\_reference {**

**publisher = "MicrosoftWindowsDesktop"**

**offer     = "Windows-10"**

**sku       = "20h2-pro"**

**version   = "latest"**

**}**

**tags = var.tags**

**}**

**resource "azurerm\_key\_vault" "dev" {**

**name                        = "kv-${var.resource\_group\_name}"**

**location                    = azurerm\_resource\_group.dev.location**

**resource\_group\_name         = azurerm\_resource\_group.dev.name**

**tenant\_id                   = var.tenant\_id**

**sku\_name                    = "standard"**

**soft\_delete\_enabled         = true**

**purge\_protection\_enabled    = true**

**network\_acls {**

**default\_action = "Deny"**

**bypass         = "AzureServices"**

**}**

**access\_policy {**

**tenant\_id = var.tenant\_id**

**object\_id = var.dev\_user\_object\_id**

**secret\_permissions = ["get", "list"]**

**}**

**tags = var.tags**

**}**

**-------------------------------------------------------------------------------------**

**3. Root Terraform Configuration**

**infra/main.tf**

**-----------------------------------------------------------------------------------**

**provider "azurerm" {**

**features {}**

**}**

**module "dev\_env" {**

**source              = "./modules/dev\_env"**

**resource\_group\_name = var.resource\_group\_name**

**location            = var.location**

**admin\_username      = var.admin\_username**

**admin\_password      = var.admin\_password**

**tenant\_id           = var.tenant\_id**

**dev\_user\_object\_id  = var.dev\_user\_object\_id**

**tags                = var.tags**

**}**

**------------------------------------------------------------------------------------**

**infra/variables.tf**

**variable "resource\_group\_name" {}**

**variable "location" {}**

**variable "admin\_username" {}**

**variable "admin\_password" {}**

**variable "tenant\_id" {}**

**variable "dev\_user\_object\_id" {}**

**variable "tags" { type = map(string) }**

**-----------------------------------------------------------------------------------**

**4. Azure DevOps Pipeline Configuration**

**azure-ci-pipeline.yaml**

**-------------------------------------------------------------------------------------**

**trigger: none**

**parameters:**

**- name: devUser**

**displayName: 'Developer Email'**

**type: string**

**stages:**

**- stage: RequestApproval**

**jobs:**

**- job: Approval**

**pool: server**

**steps:**

**- task: ManualValidation@0**

**inputs:**

**notifyUsers: 'admin@company.com'**

**instructions: 'Approve to provision a new dev environment for $(devUser)'**

**- stage: Provision**

**dependsOn: RequestApproval**

**condition: succeeded()**

**jobs:**

**- job: Terraform**

**pool:**

**vmImage: 'ubuntu-latest'**

**steps:**

**- checkout: self**

**- task: AzureKeyVault@2**

**inputs:**

**connectedServiceName: 'AzureServiceConnection'**

**keyVaultName: 'kv-devops-secrets'**

**secretsFilter: '\*'**

**- script: |**

**cd infra**

**terraform init -backend-config="storage\_account\_name=yourstorageaccount container\_name=tfstate key=dev-env.tfstate"**

**terraform plan -var="resource\_group\_name=dev-rg" -var="location=westeurope" -var="admin\_username=devadmin" -var="admin\_password=$(ADMIN\_PASSWORD)" -var="tenant\_id=$(TENANT\_ID)" -var="dev\_user\_object\_id=$(DEV\_USER\_OBJECT\_ID)" -var="tags={owner=$(devUser),env=dev,expiration=$(date -d '+7 days' +%Y-%m-%d)}"**

**terraform apply -auto-approve -var="resource\_group\_name=dev-rg" -var="location=westeurope" -var="admin\_username=devadmin" -var="admin\_password=$(ADMIN\_PASSWORD)" -var="tenant\_id=$(TENANT\_ID)" -var="dev\_user\_object\_id=$(DEV\_USER\_OBJECT\_ID)" -var="tags={owner=$(devUser),env=dev,expiration=$(date -d '+7 days' +%Y-%m-%d)}"**

**displayName: 'Terraform Apply'**

**- task: AzureCLI@2**

**displayName: 'Set Expiration Tag'**

**inputs:**

**azureSubscription: 'AzureServiceConnection'**

**scriptType: 'ps'**

**scriptLocation: 'inlineScript'**

**inlineScript: |**

**az resource tag --tags Expiration=$(date -d "+7 days" +%Y-%m-%d) --ids $(terraform output -raw vm\_id)**

**5. Security & Best Practices**

**- RBAC: Assign only required roles to the developer**

**- Key Vault: Store all secrets, restrict access**

**- Private networking: No public IPs, NSG rules**

**- Logging: Enable Log Analytics**

**- Teardown: Add a destroy pipeline/job**

**- Tags: owner, cost center, expiration**

**--------------------------------------------------------------------------------------------------------**

**6. Cost, Monitoring, and Access Controls**

**- Set Azure Budgets and cost alerts**

**- Enable auto-shutdown for VMs**

**- Use Log Analytics for monitoring**

**- Use Azure AD groups for access**

**-------------------------------------------------------------------------------------------------------**

**7. Teardown (Destroy) Pipeline**

**To safely remove all provisioned resources, add a dedicated destroy pipeline. This ensures environments are not left running and costs are controlled.**

**destroy-dev-env-pipeline.yaml**

**------------------------------------------------------------------------------------------------------------------**

**trigger: none**

**parameters:**

**- name: devUser**

**displayName: 'Developer Email'**

**type: string**

**stages:**

**- stage: RequestApproval**

**jobs:**

**- job: Approval**

**pool: server**

**steps:**

**- task: ManualValidation@0**

**inputs:**

**notifyUsers: 'admin@company.com'**

**instructions: 'Approve to destroy the dev environment for $(devUser)'**

**- stage: Destroy**

**dependsOn: RequestApproval**

**condition: succeeded()**

**jobs:**

**- job: TerraformDestroy**

**pool:**

**vmImage: 'ubuntu-latest'**

**steps:**

**- checkout: self**

**- task: AzureKeyVault@2**

**inputs:**

**connectedServiceName: 'AzureServiceConnection'**

**keyVaultName: 'kv-devops-secrets'**

**secretsFilter: '\*'**

**- script: |**

**cd infra**

**terraform init -backend-config="storage\_account\_name=yourstorageaccount container\_name=tfstate key=dev-env.tfstate"**

**terraform destroy -auto-approve -var="resource\_group\_name=dev-rg" -var="location=westeurope" -var="admin\_username=devadmin" -var="admin\_password=$(ADMIN\_PASSWORD)" -var="tenant\_id=$(TENANT\_ID)" -var="dev\_user\_object\_id=$(DEV\_USER\_OBJECT\_ID)" -var="tags={owner=$(devUser),env=dev,expiration=$(date -d '+7 days' +%Y-%m-%d)}"**

**displayName: 'Terraform Destroy'**

**--------------------------------------------------------------------------------------------------------------**

**How to use:**

**- Trigger this pipeline with the same parameters as the provisioning pipeline.**

**- Approval is required before destruction.**

**- All resources created by Terraform will be destroyed, and the state file will be updated.**

**8. Validation**

**- Test the pipeline end-to-end**

**- Validate RBAC, Key Vault, and network restrictions**

**- Check logs and cost alerts**

***\*Prepared by: Shwetha s***

***\*Date: August 23, 2025\****