

# **Azure 3-Tier 클라우드 자동화 구축**

Microsoft Cybersecurity School

**Team3**

**기준서 이승민 이헌범 임창현**

**2025.03.21**

# 목 차

<b>I. 프로젝트 개요 .....</b>	<b>2</b>
1. 선정배경	
2. 프로젝트 요구사항	
3. 개발일정	
4. 기술스택	
5. 팀원구성	
<b>II. 프로젝트 설계 .....</b>	<b>4</b>
1. Azure 아키텍처	
2. Vnet 및 서브넷 구성	
3. 공인 IP 할당	
4. 네트워크 보안그룹 적용	
5. NAT 게이트웨이 적용	
6. SSH 키 및 워드프레스 생성 시작스크립트 작성	
7. 가상머신 생성	
8. 가용성 이미지를 생성하고 VMSS 정책설정	
9. 어플리케이션 게이트웨이 적용	
10.웹서비스에 DNS 적용	
11.DB 서버 생성	
<b>III. 테라폼 구현 .....</b>	<b>11</b>
<b>IV.시행착오 및 해결방안 .....</b>	<b>30</b>
<b>V. 참고문헌 .....</b>	<b>33</b>

# I. 프로젝트 개요

## 1. 선정배경

사용자가 많은 서비스를 지원하기 위해 웹서버와 DB 서버로 구성된 2-Tier 아키텍처가 아닌 웹서버, 어플리케이션 서버 및 DB 서버로 구성된 3-Tier 아키텍처를 선택하는 것이 바람직 합니다.

3-Tier 아키텍처는 네트워크 구조가 복잡해지므로 코드형 인프라(Infrastructure as Code)를 통해 설정을 관리하는 것이 좋습니다. 따라서 이번 프로젝트에서 글로벌 퍼블릭 클라우드 Azure 에 코드형 인프라(IaC) 도구인 Terraform 을 적용하여 가용성과 보안성이 향상된 3-Tier Azure 클라우드 인프라를 구축하고자 합니다.

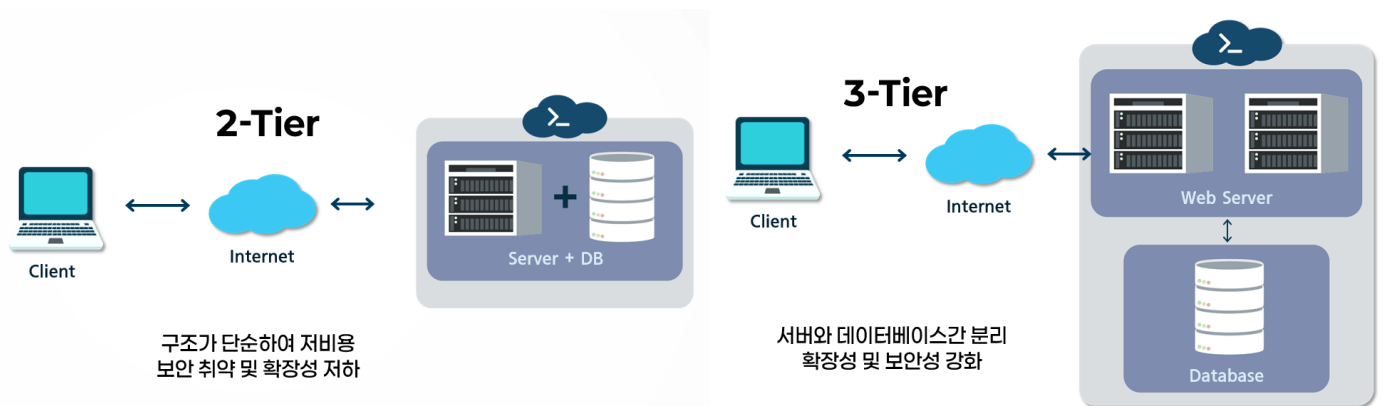


그림 1 네트워크 아키텍처 비교

## 2. 프로젝트 요구사항

### ① Bastion 을 통한 보안성 강화

Azure 네트워크 보안그룹(NSG)을 사용하여 Bastion 서버를 통해서만 Web 서버 및 DB 서버에 접속하여 관리할 수 있도록 허용하였습니다. 관리자는 SSH 개인키를 소지해야 Bastion 서버에 접속할 수 있습니다.

### ② VMSS 로 웹 서버확장성 체크

Azure VMSS(Virtual Machine Scale Set)를 사용하여 부하분산된 VM 그룹을 만들고 관리할 수 있도록 하였습니다. 설정한 서버 증감기준에 따라 VMSS 의 자동 서버확장기능으로 일시적인 사용자 증감에 대처할 수 있습니다.













### ③ 3-Tier 아키텍처 구성

외부 사용자는 로드 밸런서를 통해 Web 서버에만 접근할 수 있도록 설정하였으며, DB 서버는 내부 네트워크에서만 접근 가능하도록 제한하여 보안을 강화하였습니다.

### 3. 개발일정

일자	2025-03-17	2025-03-18	2025-03-19	2025-03-20	2025-03-21
요구사항 정의	○	○			
아키텍처 설계	○	○			
Azure 구성테스트	○	○	○	○	
테라폼 인프라자동화	○	○	○	○	
문서화 작업				○	○
수정 및 최종테스트				○	○

### 4. 기술스택

구분	Tools
Azure Services	 NAT  APP-GW  VMSS  DNS Zones
Dev tool	 Terraform 1.11.1  draw.io  Xshell 8
WEB & WAS	 Apache 2.4.63  Wordpress 6.7.2
Database	 Azure Database for MySQL servers 8.0.21
Virtual OS	 RockyLinux 9.3.20231113
Code Editor	 Visual Studio Code 1.98.2

### 5. 팀원구성

이름	기준서	이승민	이헌범	임창현
역할	아키텍처 설계 구성도 작성	요구사항 분석 업무분담	테라폼 작성 PPT 작성	최종테스트 보고서 작성

## II. 프로젝트 설계

### 1. Azure 아키텍처

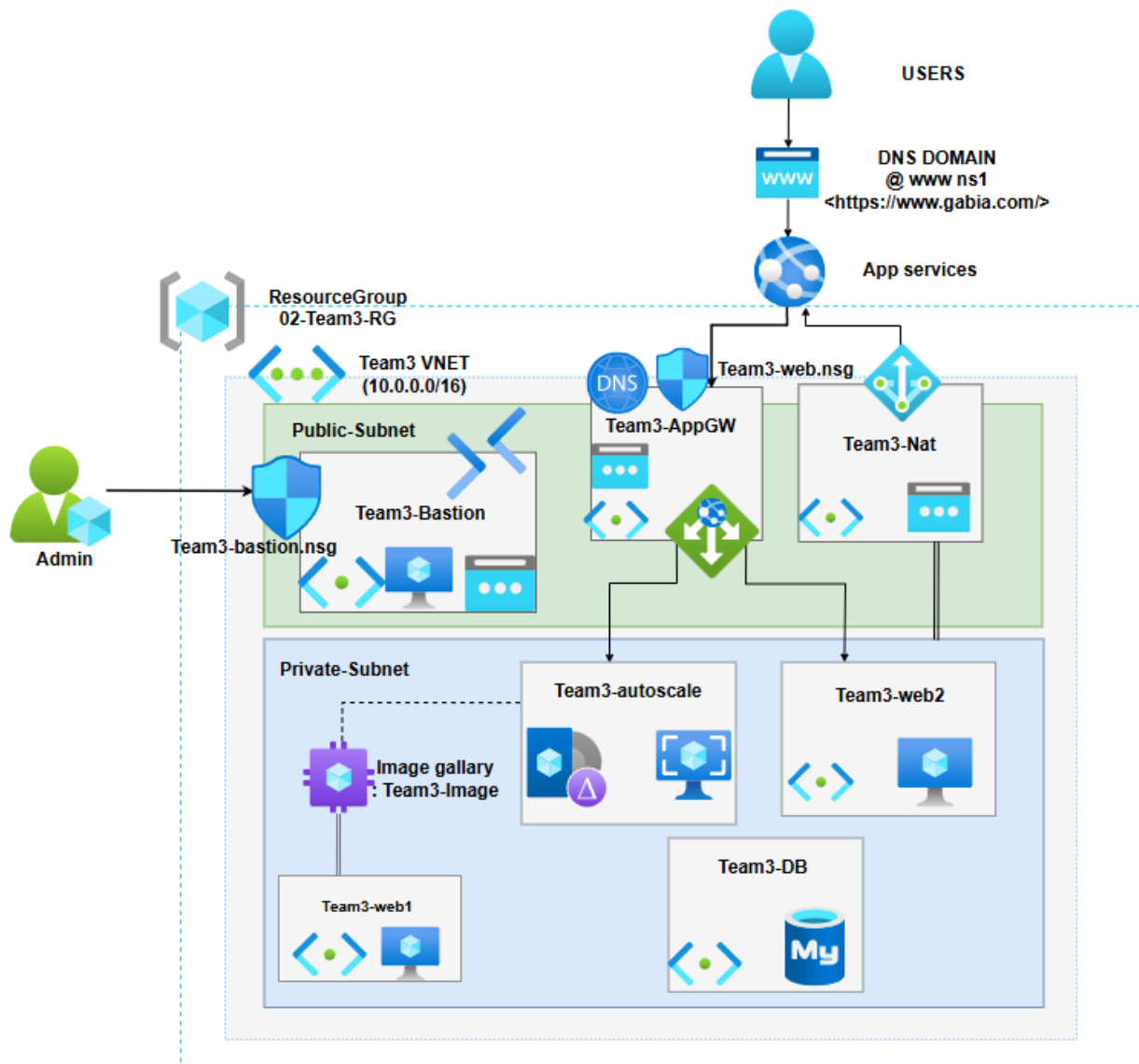


그림 2 Draw.io 에서 작성한 Azure 아키텍처

본 프로젝트는 Azure 기반 3-Tier 아키텍처 입니다.

사용자는 공개된 DNS 주소를 이용해 Public Subnet 의 어플리케이션 게이트웨이를 통하여 회사의 웹 어플리케이션을 이용할 수 있습니다. 사용자가 늘어나더라도 관리자가 설정한 정책기준에 따라 웹서버에 오토스케일링이 적용되어 이용에 차질이 없도록 설계되었습니다.

관리자는 Public Subnet 의 Bastion 가상머신을 통해 Web 서버, Web-App 서버, DB 서버를 직접 접속하여 관리할 수 있습니다. 또한 VMSS 의 Autoscaling 정책을 통해 설정한 서버의 cpu 사용률 기준점을 바탕으로 서버 갯수가 자동으로 확장 또는 축소되어 비용관리를 최적화 할 수 있습니다.

## 2. Vnet 및 서브넷 구성

### ① Vnet

Vnet	IPv4 CIDR	역할
team3_Vnet	10.0.0.0/16	Korea Central 에 위치한 Vnet

### ② Subnet

Subnet	IPv4 CIDR	역할
team3_bastion	10.0.0.0/24	보안관리자만 접속가능하다
team3_load	10.0.1.0/24	외부에서 오는 HTTP 트래픽을 로드밸런싱한다
team3_nat	10.0.2.0/24	내부 VM 이 공인 IP 을 통해 외부로 향한다
team3_web1	10.0.3.0/24	이미지화 될 WEB1 서버가 위치한 곳이다
team3_web2	10.0.4.0/24	병행운영 할 WEB2 서버가 위치한 곳이다
team3_db	10.0.5.0/24	DB 서버가 위치한 곳이다
team3_auto	10.0.6.0/24	WEB1 이미지로 생성된 서버가 오토스케일링 된다

## 3. 공인 IP 할당

IP.id	IPv4	역할
team3_bastion_ip	4.218.19.242	Bastion 의 공인 IP
team3_nat_ip	4.218.20.249	NAT 의 공인 IP
team3_appgwip	4.218.22.74	Application Gateway 의 공인 IP

```

] -> null

You can apply this plan to save these new output values to the Terraform state, without changing any
real infrastructure.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

Bastion_Public_IP = "4.218.19.242"
LB_Public_IP = "4.218.22.74"
PS C:\01_IaC\06_azure_team3>
  
```

그림 3 Terraform Output 모듈을 통해 Bastion 과 LB 의 공인 IP 를 출력

## 4. 네트워크 보안그룹 적용

### ① team3\_bastion\_nsg

관리자의 IP 에서 Bastion 서버에 SSH(22 번 포트) 접속을 허용하는 네트워크 보안그룹(NSG)입니다. NSG 를 Bastion VM 의 NIC 에 적용합니다.

### ② team3\_web\_nsg

Bastion 서버의 공인 IP 에서 WEB 서버의 SSH(22 번 포트) 접속을 허용하며, HTTP(80 번 포트)를 허용하는 네트워크 보안 그룹(NSG)입니다. NSG 를 WEB1 VM 및 WEB2 VM 의 NIC 에 적용합니다.

## 5. NAT 게이트웨이 적용

### ① team3\_natgw

내부 서브넷의 VM 이 인터넷에 직접 노출되지 않도록, NAT IP 를 이용하여 외부로 나가는 패킷을 변환합니다. 내부 서브넷 web1, web2, auto 에 적용하여 보안을 향상시킵니다.

## 6. SSH 키 및 워드프레스 생성 시작스크립트 작성

### ① local.id\_rsa

SSH 개인키를 Bastion VM 에 배치하여 WEB 서버에 SSH 접속이 가능하게 하는 스크립트입니다.

### ② local.wd

WEB1 및 WEB2 VM 에서 워드프레스를 설치하고 DB 와 연동하게 하는 스크립트입니다.

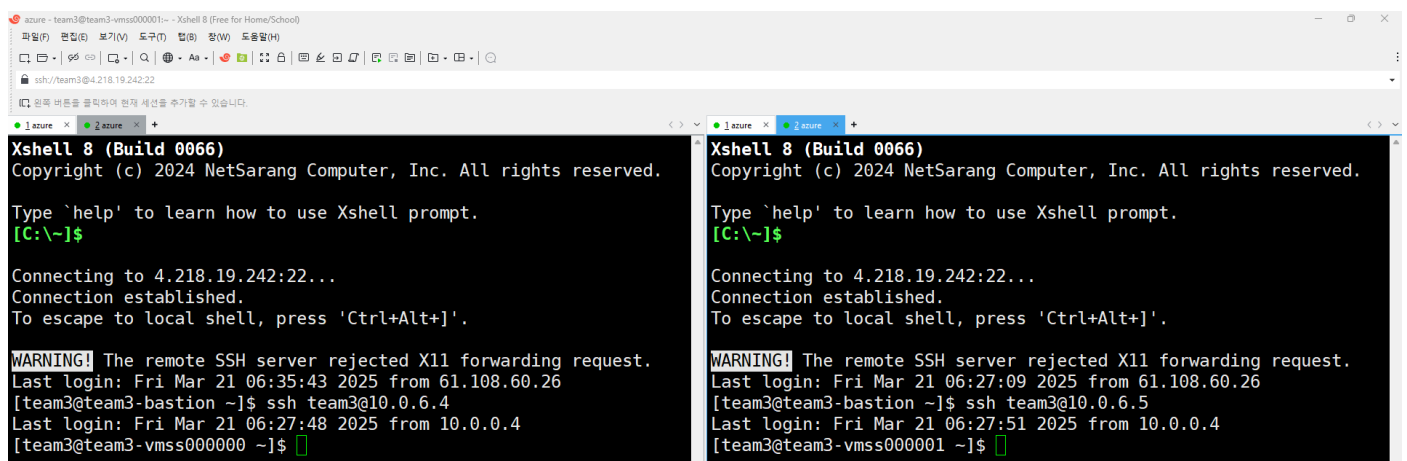


그림 4 Bastion VM 에서 WEB IMAGE VM 에 원격접속하여 관리 가능

## 7. 가상머신 생성

VM.id	IPv4	Storage	OS	역할
team3_bastion	10.0.0.4	StandardSSD	Rocky 9.3.20231113	보안관리자 PC
team3_web1	10.0.3.4	StandardSSD	Rocky 9.3.20231113	이미지 웹서버
team3_web2	10.0.4.4	StandardSSD	Rocky 9.3.20231113	병행운영 웹서버

## 8. 가용성 이미지를 생성하고 VMSS 정책설정

### ① team3\_image

WEB1 의 VM 을 이미지화하여 동일한 환경을 가진 인스턴스를 배포할 수 있도록 설정합니다. 이를 통해 일관성을 유지하면서 장애발생 시 신속하게 복구할 수 있습니다.

### ② team3\_gallery

갤러리를 생성하여 여러 VM 에서 공통으로 사용할 수 있는 이미지를 저장할 수 있도록 합니다. 갤러리를 활용하면 여러 지역의 VM 에 동일한 이미지를 배포할 수 있습니다.

### ③ team3\_shimage

갤러리에 생성된 이미지를 공유할 수 있도록 저장합니다.

### ④ team3\_version

이미지의 버전을 설정하여 업데이트 및 롤백을 관리할 수 있습니다.

### ⑤ team3\_vmss

VMSS 로 생성되는 SSD 종류와 Linux 버전 및 초기인스턴스 수를 지정할 수 있습니다. 이를 통해 자동 확장되는 VM 이 동일한 스펙과 환경을 유지할 수 있도록 관리할 수 있습니다.

### ⑥ team3\_autoscale

VMSS 의 최소 및 최대 인스턴스 개수를 지정하여 자동 확장 정책을 적용합니다. CPU 사용량을 기준으로 VM 증설과 축소를 적용하여 비용절감과 성능 최적화를 동시에 달성할 수 있습니다.



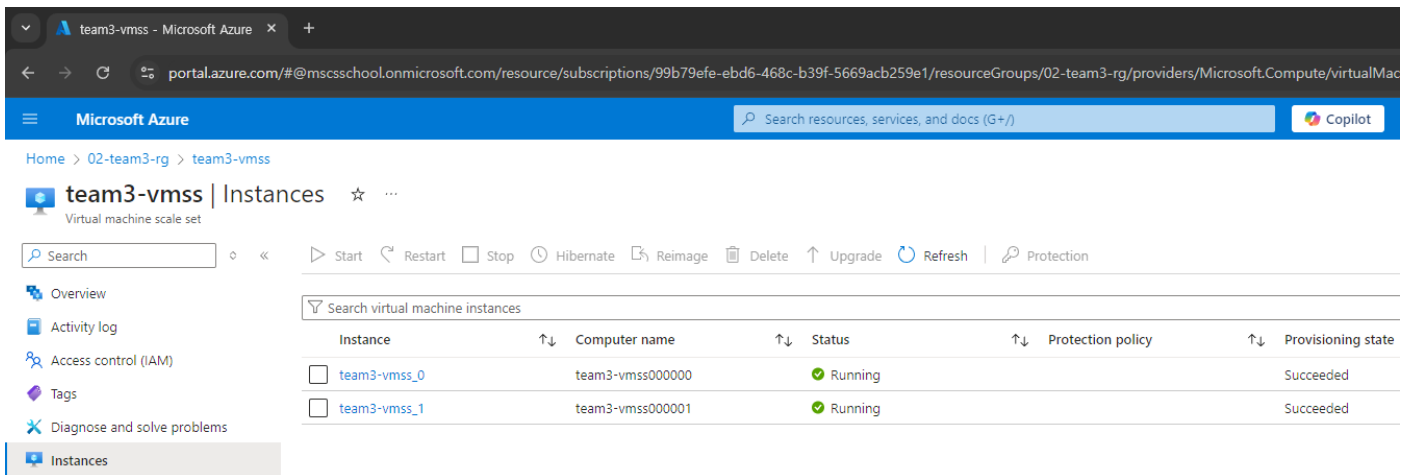


그림 5 디폴트로 설정된 두 개의 VM 이 존재

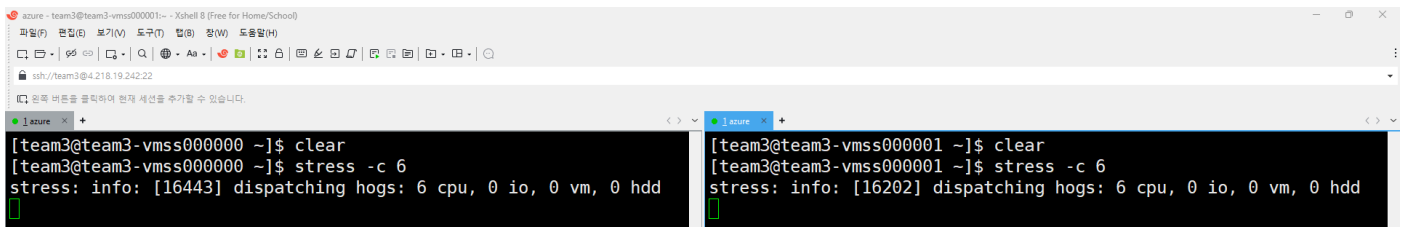


그림 6 VM 에 원격접속하여 stress 명령어로 부하테스트를 실행

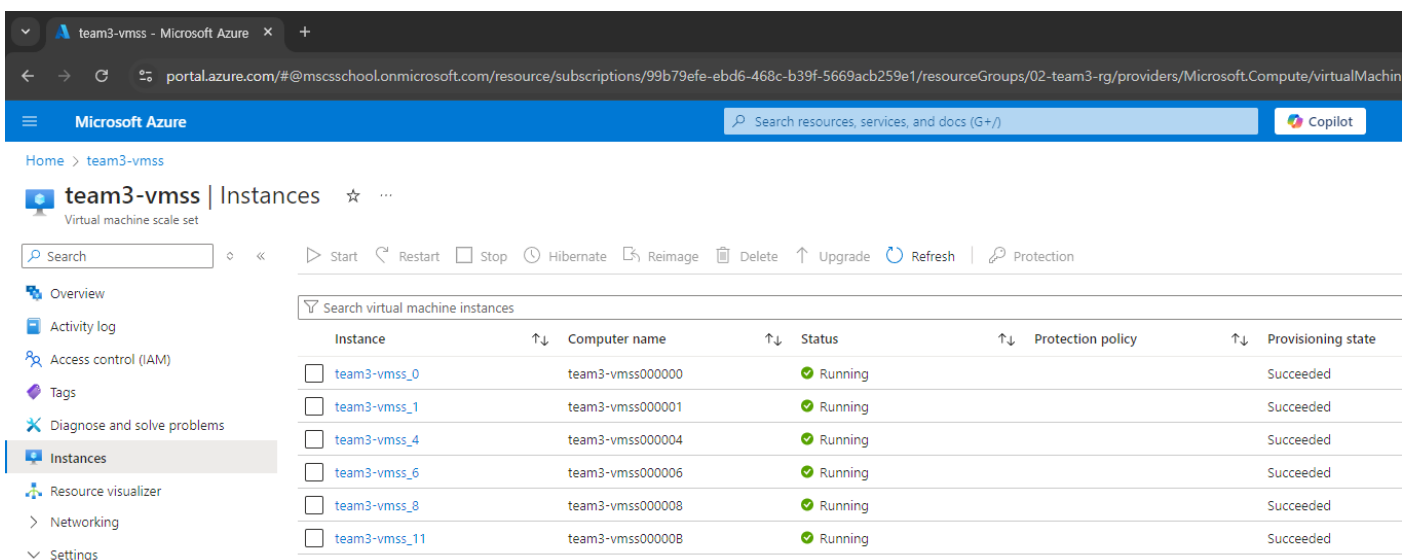


그림 7 부하테스트 실행 후 VM 이 최대설정인 6 개까지 늘어난 모습

## 9. 어플리케이션 게이트웨이 적용

### ① team3\_appgw

어플리케이션 게이트웨이는 7 계층 로드밸런서 역할을 수행합니다. 웹서비스 HTTP 에 대한 트래픽을 관리하고 특정 백엔드풀로 라우팅할 수 있도록 지정했습니다.

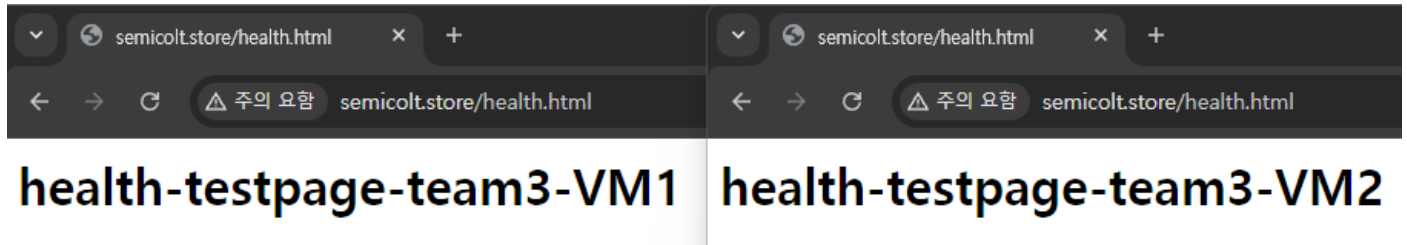


그림 8 어플리케이션 게이트웨이에 접속한 모습

## 10. 웹서비스에 DNS 적용

### ① team3\_dns

DNS 레코드를 관리하기 위한 기본 영역을 설정합니다.

### ② team3\_root\_record

루트 도메인(@)에 대해 A 레코드를 생성하여, 지정된 퍼블릭 IP 에 연결합니다.

### ③ team3\_root\_cname

www 서브도메인에 대한 A 레코드를 생성하고, 지정된 퍼블릭 IP 를 타겟으로 연결합니다.

### ④ team3\_ns

"team3-ns"라는 네임서버(NS) 레코드를 생성하여, 도메인의 네임서버를 설정합니다.

### ⑤ team3\_ptr

IP 주소에서 도메인 이름으로의 역방향 검색을 가능하게 합니다.

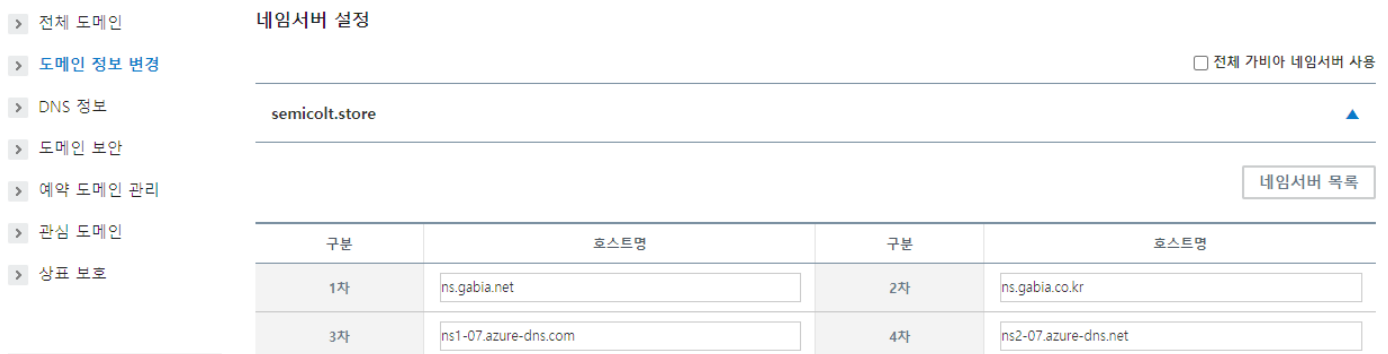


그림 9 가비아 (<https://domain.gabia.com>)에서 Azure 네임서버를 등록

## 11. DB 서버 생성

### ① team3\_pridns

Private DNS Zone 을 활용하여 MySQL 서버의 사설 도메인 네임을 관리할 수 있습니다. MySQL 서버가 내부 네트워크(VNet)에서만 접근 가능하도록 Private Link 를 설정하는 경우, Private DNS 를 함께 설정해야 내부에서 FQDN(예: mysql.team3.private.azure)으로 접근이 가능합니다.

### ② team3\_dns\_link

Private DNS Zone 을 특정 VNet 에 연결하는 역할을 합니다. Azure 의 Private DNS Zone 은 기본적으로 네트워크와 연결되어 있지 않으므로, VNet 에서 해당 DNS 를 사용할 수 있도록 Virtual Network Link 를 설정해야 합니다. 이를 통해 동일한 VNet 또는 피어링된 VNet 내에서 MySQL 서버를 도메인 네임을 통해 접근할 수 있습니다.

### ③ team3\_mysql

Azure MySQL 서버의 이름 또는 Terraform 에서 정의한 논리적 식별자를 의미합니다. Terraform 코드에서 MySQL 서버의 리소스를 생성할 때 Identifier 을 설정하는 데 사용됩니다.

### ④ team3\_mysql\_ep

MySQL 서버가 생성된 후, 클라이언트가 접속할 수 있도록 제공되는 엔드포인트(FQDN 또는 IP 주소)를 의미합니다. Azure MySQL 은 기본적으로 Public Endpoint 를 제공하지만, Private Link 를 설정한 경우 내부 VNet 을 통해 Private Endpoint 로 접속해야 합니다.

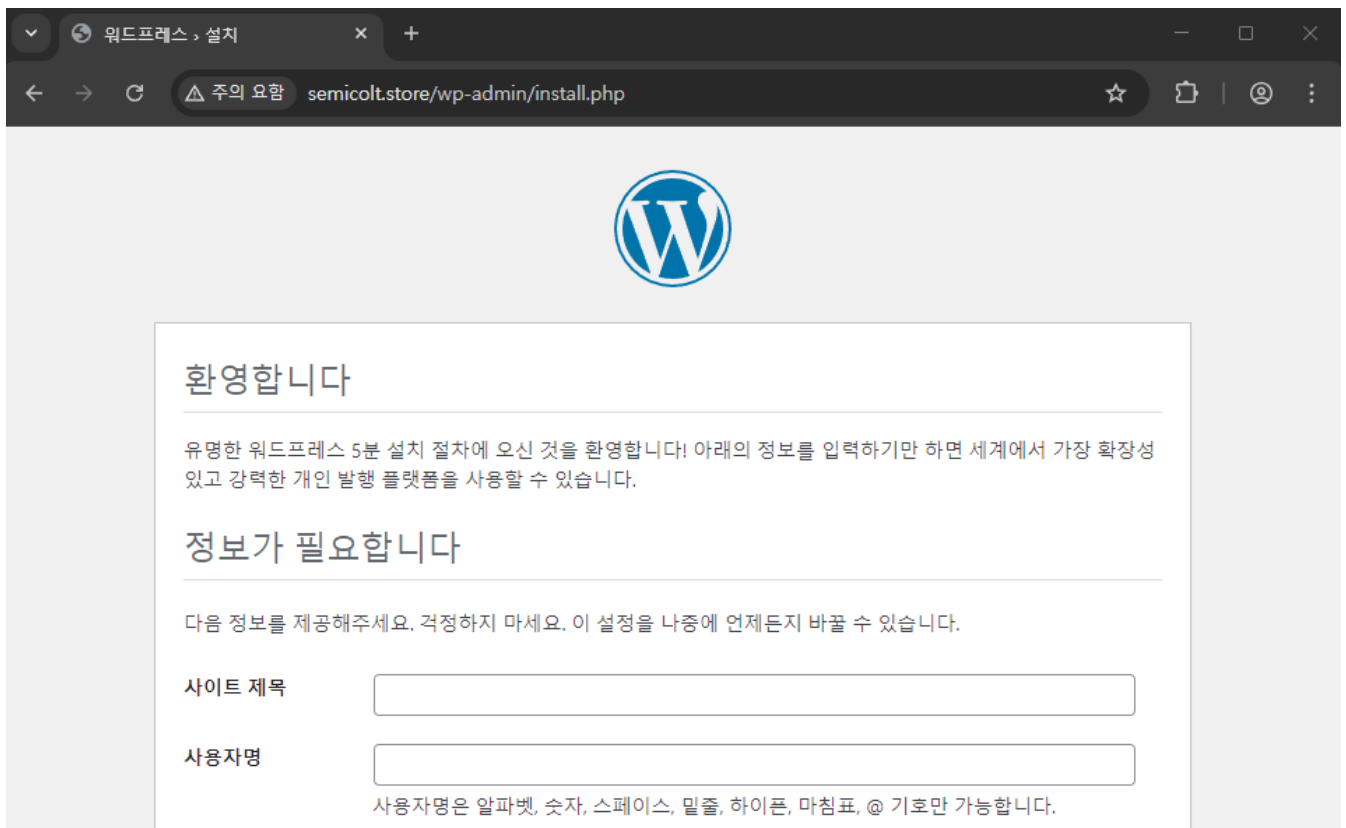


그림 10 DNS 로 접속한 워드프레스

### III. 테라폼 구현

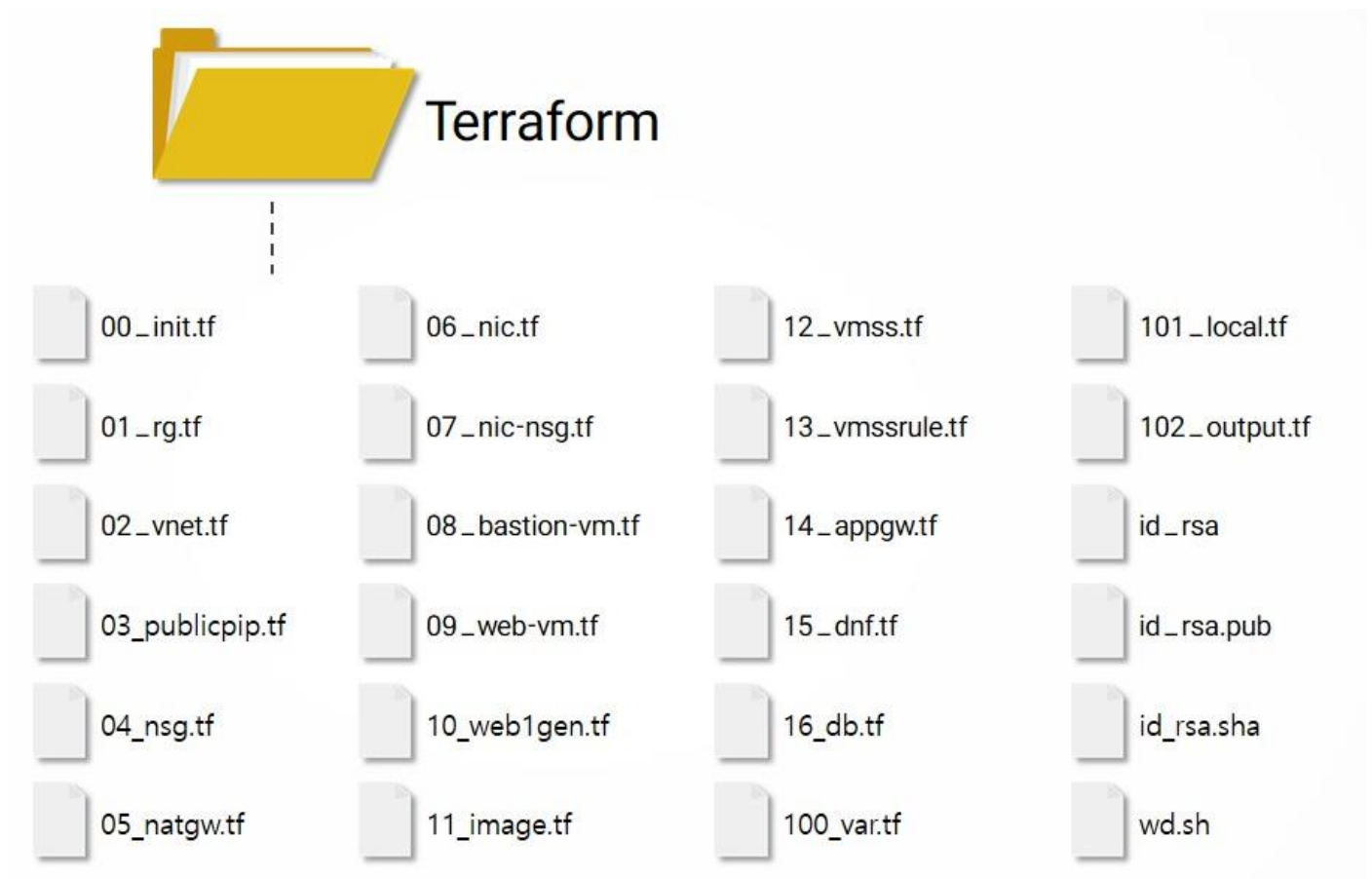


그림 11 환경구성을 위한 테라폼 파일

#### 00.init.tf

```
terraform {
  required_providers {
    azurerm = {
      source = "hashicorp/azurerm"
      version = "~> 4.23"
    }
  }
}

provider "azurerm" {
  subscription_id = var.subid
  features {}
}
```

## 01\_rg.tf

```
# Create Resource Group
resource "azurerm_resource_group" "team3_rg" {
  name      = "02-${var.name}-rg"
  location  = var.location
}
```

## 02\_vnet.tf

```
# Create Virtual Network
resource "azurerm_virtual_network" "team3_vnet" {
  name                = "${var.name}-vnet"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  address_space       = ["10.0.0.0/16"]
}

# Create Public Subnet for Bastion
resource "azurerm_subnet" "team3_bastion" {
  name                = "${var.name}-bastion"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.0.0/24"]
}

# Create Load Subnet
resource "azurerm_subnet" "team3_load" {
  name                = "${var.name}-load"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.1.0/24"]
}

# Create NAT Subnet
resource "azurerm_subnet" "team3_nat" {
  name                = "${var.name}-nat"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.2.0/24"]
}

// Web Subnet 들이 필요없는 상황이라 삭제 고려
# Create Web1 Subnet
resource "azurerm_subnet" "team3_web1" {
  name                = "${var.name}-web1"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.3.0/24"]
}
```

```

# Create Web2 Subnet
resource "azurerm_subnet" "team3_web2" {
  name                = "${var.name}-web2"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.4.0/24"]
}

# Create DB Subnet
resource "azurerm_subnet" "team3_db" {
  name                = "${var.name}-db"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.5.0/24"]
}

# Subnet for Auto Scale
resource "azurerm_subnet" "team3_auto" {
  name                = "${var.name}-auto"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.6.0/24"]
}

```

### 03\_publicip.tf

```

# Public IP for Bastion VM
resource "azurerm_public_ip" "team3_bastion_ip" {
  name                = "${var.name}-bastion-ip"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  allocation_method   = "Static"
  sku                 = "Standard"
}

# Public IP for NAT Gateway
resource "azurerm_public_ip" "team3_nat_ip" {
  name                = "${var.name}-nat-ip"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  allocation_method   = "Static"
  sku                 = "Standard"
}

# Public IP for Load Balancer
resource "azurerm_public_ip" "team3_appgwip" {
  name                = "${var.name}-lb-ip"
  resource_group_name = azurerm_resource_group.team3_rg.name
}

```

```

location          = azurerm_resource_group.team3_rg.location
allocation_method = "Static"
sku               = "Standard"
}

```

## 04\_nsg.tf

*# Network Security Group for Bastion*

```

resource "azurerm_network_security_group" "team3_bastion_nsg" {
  name          = "${var.name}-bat-nsg"
  location      = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name
  # Bastion SSH Allow Rule
  security_rule {
    name          = "Allow-SSH-From-Host"
    priority      = 200
    direction     = "Inbound"
    access        = "Allow"
    protocol      = "Tcp"
    source_port_range = "*"
    destination_port_range = "22"
    source_address_prefix = var.local_public_ip
    destination_address_prefix = var.bastion_ip
  }
}

```

*# Network Security Group for Internal Network*

```

resource "azurerm_network_security_group" "team3_web_nsg" {
  name          = "${var.name}-web-nsg"
  location      = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name
  # Web SSH Allow Rule
  security_rule {
    name          = "Allow-SSH-From-Bastion"
    priority      = 200
    direction     = "Inbound"
    access        = "Allow"
    protocol      = "Tcp"
    source_port_range = "*"
    destination_port_range = "22"
    source_address_prefix = var.bastion_ip
    destination_address_prefix = "*"
  }
  # Web Http Allow Rule
  security_rule {
    name          = "Allow-HTTP-From-All"
    priority      = 210
    direction     = "Inbound"

```

```

    access                = "Allow"
    protocol               = "Tcp"
    source_port_range      = "*"
    destination_port_range = "80"
    source_address_prefix  = "*"
    destination_address_prefix = var.auto_ip
  }
}

```

## 05\_natgw.tf

```

# Create NAT Gateway
resource "azurerm_nat_gateway" "team3_natgw" {
  name                = "${var.name}-natgw"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name
}

# NAT Gateway Association

resource "azurerm_subnet_nat_gateway_association" "team3_web1_nat" {
  subnet_id          = azurerm_subnet.team3_web1.id
  nat_gateway_id     = azurerm_nat_gateway.team3_natgw.id
}

resource "azurerm_subnet_nat_gateway_association" "team3_web2_nat" {
  subnet_id          = azurerm_subnet.team3_web2.id
  nat_gateway_id     = azurerm_nat_gateway.team3_natgw.id
}

resource "azurerm_subnet_nat_gateway_association" "team3_web_nat" {
  subnet_id          = azurerm_subnet.team3_auto.id
  nat_gateway_id     = azurerm_nat_gateway.team3_natgw.id
}

# Attachment Public IP -> NAT Gateway
resource "azurerm_nat_gateway_public_ip_association" "team3_natgwp_pubip" {
  nat_gateway_id      = azurerm_nat_gateway.team3_natgw.id
  public_ip_address_id = azurerm_public_ip.team3_nat_ip.id
}

```

## 06\_nic.tf

```

# Network Interface Card for Bastion VM
resource "azurerm_network_interface" "team3_bat_nic" {
  name                = "${var.name}-bat-nic"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name
}

```



```

ip_configuration {
  name                = "${var.name}-bat-ip"
  subnet_id           = azurerm_subnet.team3_bastion.id
  private_ip_address_allocation = "Static"
  private_ip_address   = var.bastion_ip
  public_ip_address_id = azurerm_public_ip.team3_bastion_ip.id
}
}

```

*# Network Interface Card for Web1*

```

resource "azurerm_network_interface" "team3_web1_nic" {
  name                = "${var.name}-web1-nic"
  location             = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name

  ip_configuration {
    name                = "${var.name}-web1-ip"
    subnet_id           = azurerm_subnet.team3_web1.id
    private_ip_address_allocation = "Static"
    private_ip_address   = "10.0.3.4"
  }
}

```

*# Network Interface Card for Web2*

```

resource "azurerm_network_interface" "team3_web2_nic" {
  name                = "${var.name}-web2-nic"
  location             = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name

  ip_configuration {
    name                = "${var.name}-web2-ip"
    subnet_id           = azurerm_subnet.team3_web2.id
    private_ip_address_allocation = "Static"
    private_ip_address   = "10.0.4.4"
  }
}

```

## 07\_nsg-nic.tf

*# Bastion NIC <-> NSG*

```

resource "azurerm_network_interface_security_group_association"
"team3_bat_nic_nsgasso" {
  network_interface_id      = azurerm_network_interface.team3_bat_nic.id
  network_security_group_id = azurerm_network_security_group.team3_bastion_nsg.id
}

```

*# Web NIC <-> NSG*

```

resource "azurerm_network_interface_security_group_association"
"team3_web1_nic_nsgasso" {
  network_interface_id      = azurerm_network_interface.team3_web1_nic.id
  network_security_group_id = azurerm_network_security_group.team3_web_nsg.id
}

resource "azurerm_network_interface_security_group_association"
"team3_web2_nic_nsgasso" {
  network_interface_id      = azurerm_network_interface.team3_web2_nic.id
  network_security_group_id = azurerm_network_security_group.team3_web_nsg.id
}

```

## 08\_bastion\_vm.tf

```

# Create Bastion VM
resource "azurerm_linux_virtual_machine" "team3_bastion" {
  name                  = "${var.name}-bastion"
  resource_group_name  = azurerm_resource_group.team3_rg.name
  location              = azurerm_resource_group.team3_rg.location
  size                 = "Standard_F1s"
  admin_username       = var.name
  network_interface_ids = [azurerm_network_interface.team3_bat_nic.id]
  user_data             = base64encode(local.id_rsa)

  admin_ssh_key {
    username   = var.name
    public_key = file("id_rsa.pub")
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "StandardSSD_LRS"
  }

  source_image_reference {
    publisher = "resf"
    offer     = "rockylinux-x86_64"
    sku       = "9-lvm"
    version   = "9.3.20231113"
  }

  plan {
    publisher = "resf"
    product   = "rockylinux-x86_64"
    name      = "9-lvm"
  }
}

```

## 09\_web\_vm.tf

*# Create VM Web1*

```
resource "azurerm_linux_virtual_machine" "team3_web1" {
  name                = "${var.name}-web1"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  size               = "Standard_F1s"
  admin_username     = var.name
  network_interface_ids = [azurerm_network_interface.team3_web1_nic.id]
  user_data           = base64encode(local.wdimage)

  admin_ssh_key {
    username   = var.name
    public_key = file("id_rsa.pub")
  }

  os_disk {
    caching          = "ReadWrite"
    storage_account_type = "StandardSSD_LRS"
  }

  source_image_reference {
    publisher = "resf"
    offer     = "rockylinux-x86_64"
    sku       = "9-lvm"
    version   = "9.3.20231113"
  }

  plan {
    publisher = "resf"
    product   = "rockylinux-x86_64"
    name      = "9-lvm"
  }
}
```

*# Create VM Web2*

```
resource "azurerm_linux_virtual_machine" "team3_web2" {
  name                = "${var.name}-web2"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  size               = "Standard_F1s"
  admin_username     = var.name
  network_interface_ids = [azurerm_network_interface.team3_web2_nic.id]
  user_data           = base64encode(local.wd)

  admin_ssh_key {
    username   = var.name
    public_key = file("id_rsa.pub")
  }
}
```

```

}

os_disk {
  caching          = "ReadWrite"
  storage_account_type = "StandardSSD_LRS"
}

source_image_reference {
  publisher = "resf"
  offer     = "rockylinux-x86_64"
  sku       = "9-lvm"
  version   = "9.3.20231113"
}

plan {
  publisher = "resf"
  product   = "rockylinux-x86_64"
  name      = "9-lvm"
}
}

```

## 10\_web1gen.tf

```

resource "time_sleep" "wait_before_stop" {

  create_duration = "120s"

  depends_on = [azurerm_linux_virtual_machine.team3_web1]
}

resource "null_resource" "stop_web1" {
  provisioner "local-exec" {
    command = "az vm stop --resource-group 02-team3-rg --name team3-web1"
  }

  depends_on = [time_sleep.wait_before_stop]
}

resource "null_resource" "deal_web1" {
  provisioner "local-exec" {
    command = "az vm deallocate --resource-group 02-team3-rg --name team3-web1"
  }

  depends_on = [null_resource.stop_web1]
}

resource "null_resource" "gen_web1" {
  provisioner "local-exec" {
    command = "az vm generalize --resource-group 02-team3-rg --name team3-web1"
  }
}

```

```

}

depends_on = [null_resource.deal_web1]
}

```

## 11\_image.tf

```

resource "azurerm_image" "team3_image" {
  name                        = "${var.name}-image"
  resource_group_name        = azurerm_resource_group.team3_rg.name
  location                   = azurerm_resource_group.team3_rg.location
  source_virtual_machine_id = azurerm_linux_virtual_machine.team3_web1.id
  hyper_v_generation         = "V2"

  depends_on = [null_resource.gen_web1]
}

resource "azurerm_shared_image_gallery" "team3_gallery" {
  name                        = "${var.name}gallery"
  resource_group_name        = azurerm_resource_group.team3_rg.name
  location                   = azurerm_resource_group.team3_rg.location

  depends_on = [azurerm_image.team3_image]
}

resource "azurerm_shared_image" "team3_shimage" {
  name                        = "${var.name}-shimage"
  gallery_name               = azurerm_shared_image_gallery.team3_gallery.name
  resource_group_name        = azurerm_resource_group.team3_rg.name
  location                   = azurerm_resource_group.team3_rg.location
  os_type                    = "Linux"
  specialized                 = false
  hyper_v_generation         = "V2"

  identifier {
    publisher = var.name
    offer     = "web-template"
    sku       = "wordpress-v1"
  }

  depends_on = [azurerm_shared_image_gallery.team3_gallery]
}

resource "azurerm_shared_image_version" "team3_version" {
  name                        = "1.0.0"
  gallery_name               = azurerm_shared_image_gallery.team3_gallery.name
  image_name                 = azurerm_shared_image.team3_shimage.name
  resource_group_name        = azurerm_resource_group.team3_rg.name
}

```

```

location          = azurerm_resource_group.team3_rg.location
managed_image_id  = azurerm_image.team3_image.id

target_region {
  name              = azurerm_shared_image.team3_shimage.location
  regional_replica_count = 6
}

depends_on = [azurerm_shared_image.team3_shimage]
}

```

## 12\_vmss.tf

```

# Create VMSS
resource "azurerm_linux_virtual_machine_scale_set" "team3_vmss" {
  name                = "${var.name}-vmss"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  upgrade_mode        = "Manual"
  sku                 = "Standard_F1s"
  instances            = 2
  admin_username      = var.name

  source_image_id = azurerm_shared_image_version.team3_version.id

  plan {
    name       = "9-lvm"
    publisher  = "resf"
    product    = "rockylinux-x86_64"
  }

  admin_ssh_key {
    username   = var.name
    public_key = file("id_rsa.pub")
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "StandardSSD_LRS"
  }

  network_interface {
    name      = "${var.name}-vmss-nic"
    primary  = true

    ip_configuration {
      name          = "${var.name}-nic"
      primary       = true

```

```

        subnet_id = azurerm_subnet.team3_auto.id
        application_gateway_backend_address_pool_ids =
[one(azurerm_application_gateway.team3_appgw.backend_address_pool[*].id)]
    }
}

depends_on = [azurerm_shared_image_version.team3_version]
}

```

### 13\_vmssrule.tf

```

resource "azurerm_monitor_autoscale_setting" "team3_autoscale" {
  name = "${var.name}-autoscale"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location = azurerm_resource_group.team3_rg.location
  target_resource_id = azurerm_linux_virtual_machine_scale_set.team3_vmss.id

  profile {
    name = "${var.name}-Profile"

    capacity {
      default = 2
      minimum = 2
      maximum = 6
    }

    rule {
      metric_trigger {
        metric_name = "Percentage CPU"
        metric_resource_id = azurerm_linux_virtual_machine_scale_set.team3_vmss.id
        time_grain = "PT1M"
        statistic = "Average"
        time_window = "PT5M"
        time_aggregation = "Average"
        operator = "GreaterThan"
        threshold = 75
        metric_namespace = "microsoft.compute/virtualmachinescalesets"
      }

      scale_action {
        direction = "Increase"
        type = "ChangeCount"
        value = "1"
        cooldown = "PT1M"
      }
    }
  }

  rule {
    metric_trigger {

```

```

        metric_name      = "Percentage CPU"
        metric_resource_id = azurerm_linux_virtual_machine_scale_set.team3_vmss.id
        time_grain        = "PT1M"
        statistic         = "Average"
        time_window       = "PT5M"
        time_aggregation  = "Average"
        operator          = "LessThan"
        threshold         = 25
    }

    scale_action {
        direction = "Decrease"
        type      = "ChangeCount"
        value     = "1"
        cooldown  = "PT1M"
    }
}
}
}
}

```

## 14\_appgw.tf

```

# Create Application Gateway
resource "azurerm_application_gateway" "team3_appgw" {
    name                = "${var.name}-appgw"
    resource_group_name = azurerm_resource_group.team3_rg.name
    location            = azurerm_resource_group.team3_rg.location

    sku {
        name      = "Basic"
        tier       = "Basic"
        capacity = 2
    }

    gateway_ip_configuration {
        name      = "${var.name}-gateway-ip-configuration"
        subnet_id = azurerm_subnet.team3_load.id
    }

    # For HTTP Service
    frontend_port {
        name = var.frontend_port_name
        port = 80
    }

    # Attachment Public IP
    frontend_ip_configuration {
        name                = var.frontend_ip_configuration_name
        public_ip_address_id = azurerm_public_ip.team3_appgwip.id
    }
}

```



```

# Backend IP Setting
backend_address_pool {
  name          = var.backend_address_pool_name
  ip_addresses  = ["10.0.4.4"]
}

backend_http_settings {
  name              = var.http_setting_name
  cookie_based_affinity = "Disabled"
  path              = "/"
  port              = 80
  protocol           = "Http"
  request_timeout    = 60
}

http_listener {
  name                  = var.listener_name
  frontend_ip_configuration_name = var.frontend_ip_configuration_name
  frontend_port_name     = var.frontend_port_name
  protocol                = "Http"
}

request_routing_rule {
  name                  = var.request_routing_rule_name
  rule_type             = "Basic"
  http_listener_name     = var.listener_name
  backend_address_pool_name = var.backend_address_pool_name
  backend_http_settings_name = var.http_setting_name
  priority               = 100
}
}

```

## 15\_dns.tf

```

resource "azurerm_dns_zone" "team3_dns" {
  name          = "semicolt.store"
  resource_group_name = azurerm_resource_group.team3_rg.name
}

resource "azurerm_dns_a_record" "team3_root_record" {
  name          = "@"
  resource_group_name = azurerm_resource_group.team3_rg.name
  zone_name     = azurerm_dns_zone.team3_dns.name
  ttl           = 300
  target_resource_id = azurerm_public_ip.team3_appgwip.id
}

resource "azurerm_dns_a_record" "team3_root_cname" {
  name          = "www"
  resource_group_name = azurerm_resource_group.team3_rg.name
}

```

```

zone_name      = azurerm_dns_zone.team3_dns.name
ttl            = 300
target_resource_id = azurerm_public_ip.team3_appgwip.id
}

resource "azurerm_dns_ns_record" "team3_ns" {
  name          = "team3-ns"
  zone_name     = azurerm_dns_zone.team3_dns.name
  resource_group_name = azurerm_resource_group.team3_rg.name
  ttl          = 300
  records      = ["ns1.semicolt.store"]
}

resource "azurerm_dns_ptr_record" "team3_ptr" {
  name          = "team3-ptr"
  zone_name     = azurerm_dns_zone.team3_dns.name
  resource_group_name = azurerm_resource_group.team3_rg.name
  ttl          = 300
  records      = ["semicolt.store"]
}

```

## 16\_db.tf

```

# DNS Zone for MySQL
resource "azurerm_private_dns_zone" "team3_pridns" {
  name          = "${var.name}.mysql.database.azure.com"
  resource_group_name = azurerm_resource_group.team3_rg.name
}

# DNS zone Virtual Network Links
resource "azurerm_private_dns_zone_virtual_network_link" "team3_dns_link" {
  name          = "${var.name}-pridns-vnetzone.com"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_id = azurerm_virtual_network.team3_vnet.id
  private_dns_zone_name = azurerm_private_dns_zone.team3_pridns.name
}

# Create MySQL Server
resource "azurerm_mysql_flexible_server" "team3_mysql" {
  name          = "${var.name}-mysql"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location      = azurerm_resource_group.team3_rg.location
  administrator_login = var.name
  administrator_password = var.password
  backup_retention_days = 7
  geo_redundant_backup_enabled = false
  sku_name      = "B_Standard_B1ms"
  version       = "8.0.21"
}

```

```

lifecycle {
  ignore_changes = [
    zone,
  ]
}
}

# Create Private Endpoint
resource "azurerm_private_endpoint" "team3_mysql_ep" {
  name                = "${var.name}-mysql-ep"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location             = azurerm_resource_group.team3_rg.location
  subnet_id           = azurerm_subnet.team3_db.id

  private_service_connection {
    name                        = "mysql"
    private_connection_resource_id = azurerm_mysql_flexible_server.team3_mysql.id
    subresource_names          = ["mysqlServer"]
    is_manual_connection       = false
  }

  private_dns_zone_group {
    name                = "${var.name}-mysql-dns-zone-group"
    private_dns_zone_ids = [azurerm_private_dns_zone.team3_pridns.id]
  }
}

# Create Database Name "wordpress"
resource "azurerm_mysql_flexible_database" "team3_db" {
  charset          = "utf8mb4"
  collation         = "utf8mb4_unicode_ci"
  name              = "wordpress"
  resource_group_name = azurerm_resource_group.team3_rg.name
  server_name       = azurerm_mysql_flexible_server.team3_mysql.name
}

# Change Parameter
resource "azurerm_mysql_flexible_server_configuration" "team3_mysql_config" {
  name                = "require_secure_transport"
  resource_group_name = azurerm_resource_group.team3_rg.name
  server_name         = azurerm_mysql_flexible_server.team3_mysql.name
  value               = "OFF"
}

```

## 100\_var.tf

```

variable "location" {
  type    = string

```

```

    default = "Korea Central"
}

variable "name" {
    type    = string
    default = "team3"
}

variable "bastion_ip" {
    type    = string
    default = "10.0.0.4"
}

variable "local_public_ip" {
    type    = string
    default = "61.108.60.26"
}

variable "auto_ip" {
    type    = string
    default = "10.0.6.0/24"
}

variable "password" {
    type    = string
    default = "It12345!"
}

variable "id_rsa" {
    type    = string
    default = "-----BEGIN RSA PRIVATE KEY-----
\nMIIIEogIBAAKCAQEak8ugubI2AOE5utL6qcZ68khg5KsU6BnNRzQF40MasdupYIRh\nwNLKT+kT/TJhdOsrEd
(blank)
DWQ=\n-----END RSA PRIVATE KEY-----"
}

variable "subid" {
    type    = string
    default = "99b79efe-ebd6-468c-b39f-5669acb259e1"
}

variable "backend_address_pool_name" {
    default = "team3-Backaddrpool"
}

variable "frontend_port_name" {
    default = "team3-FrontendPort"
}

```

```

variable "frontend_ip_configuration_name" {
  default = "team3-AGIPConfig"
}

variable "http_setting_name" {
  default = "team3-HTTPsetting"
}

variable "listener_name" {
  default = "team3-Listener"
}

variable "request_routing_rule_name" {
  default = "team3-RoutingRule"
}

```

## 101\_local.tf

```

# USER DATA File for Bastion VM
locals {
  id_rsa = <<USER_DATA
#!/bin/bash
mkdir /home/${var.name}/.ssh
echo -e "${var.id_rsa}" > /home/${var.name}/.ssh/id_rsa
chmod 600 /home/${var.name}/.ssh/id_rsa
chown ${var.name}.${var.name} /home/${var.name}/.ssh/id_rsa
USER_DATA
}

# USER DATA File for Web Service VM
locals {
  wd = <<USER_DATA
#!/bin/bash
setenforce 0
grubby --update-kernel ALL --args selinux=0
yum install -y httpd wget tar php php-cli php-pdo php-fpm php-json php-mysqlnd
wget https://ko.wordpress.org/wordpress-6.7.2-ko_KR.tar.gz
tar xvfz wordpress-6.7.2-ko_KR.tar.gz
cp -ar wordpress/* /var/www/html/
sed -i "s/DirectoryIndex index.html/DirectoryIndex index.php/g"
/etc/httpd/conf/httpd.conf
cp /var/www/html/{wp-config-sample.php,wp-config.php}
sed -i "s/database_name_here/wordpress/g" /var/www/html/wp-config.php
sed -i "s/username_here/${var.name}/g" /var/www/html/wp-config.php
sed -i "s/password_here/It12345!/g" /var/www/html/wp-config.php
sed -i "s/localhost/10.0.5.4/g" /var/www/html/wp-config.php
cat > /var/www/html/health.html << eof

```

```

<html><body><h1>health-testpage</h1></body></html>
eof
chown -R apache.apache /var/www
systemctl enable --now httpd
USER_DATA
}

# USER DATA File for Web Service VM
locals {
  wdimage = <<USER_DATA
#!/bin/bash
setenforce 0
grubby --update-kernel ALL --args selinux=0
yum install -y WALinuxAgent httpd wget tar php php-cli php-pdo php-fpm php-json php-
mysqlnd
wget https://ko.wordpress.org/wordpress-6.7.2-ko_KR.tar.gz
tar xvfz wordpress-6.7.2-ko_KR.tar.gz
cp -ar wordpress/* /var/www/html/
sed -i "s/DirectoryIndex index.html/DirectoryIndex index.php/g"
/etc/httpd/conf/httpd.conf
cp /var/www/html/{wp-config-sample.php,wp-config.php}
sed -i "s/database_name_here/wordpress/g" /var/www/html/wp-config.php
sed -i "s/username_here/${var.name}/g" /var/www/html/wp-config.php
sed -i "s/password_here/It12345!/g" /var/www/html/wp-config.php
sed -i "s/localhost/10.0.5.4/g" /var/www/html/wp-config.php
cat > /var/www/html/health.html << eof
<html><body><h1>health-testpage</h1></body></html>
eof
chown -R apache.apache /var/www
systemctl enable --now httpd
waagent -deprovision -force
USER_DATA
}

```

102\_output.tf

```

# Check Public IP
output "Bastion_Public_IP" {
  value = azurerm_public_ip.team3_bastion_ip.ip_address
}

output "LB_Public_IP" {
  value = azurerm_public_ip.team3_appgwip.ip_address
}

```

## IV. 시행착오 및 해결방안

### ➤ 불충분한 RockyLinux 이미지 정보

VM에 사용할 Rockylinux를 정보를 확인하려고 했으나 Azure 마켓플레이스에서는 코드에 작성할 수 있을만큼 충분한 정보가 노출되어 있지 않았습니다. 웹 검색결과를 참고하여 아래와 같이 `az vm image list` 명령어를 통해 얻은 버전정보로 코드를 작성할 수 있었습니다.



**Rocky Linux for x86\_64 (AMD64) - Official** Remove from Favorites

The Rocky Enterprise Software Foundation | Virtual Machine

★ 3.0 (5 ratings)

Overview Plans + Pricing **Usage Information + Support** Ratings + Reviews

**Usage Information**

Publisher ID  
resf

Product ID  
rockylinux-x86\_64

Plan ID ⓘ  
8-lvm

**Useful Links**

Forums  
Chat

**Support**

Support

그림 12 Azure Portal에서 확인되는 RockyLinux 이미지 정보

```
C:\Users\secu16>az vm image list --publisher resf --offer rockylinux-x86_64 --all --output table
```

Architecture	Offer	Publisher	Skus	Urn	Version
x64	rockylinux-x86_64	resf	8-base	resf:rockylinux-x86_64:8-base:8.9.20231119	8.9.20231119
x64	rockylinux-x86_64	resf	8-lvm	resf:rockylinux-x86_64:8-lvm:8.9.20231119	8.9.20231119
x64	rockylinux-x86_64	resf	9-base	resf:rockylinux-x86_64:9-base:9.3.20231113	9.3.20231113
x64	rockylinux-x86_64	resf	9-lvm	resf:rockylinux-x86_64:9-lvm:9.3.20231113	9.3.20231113

그림 13 az vm image list 명령어로 버전 확인

### ✓ 해결방안을 적용한 09\_web\_vm.tf

```
source_image_reference {  
  publisher = "resf"  
  offer     = "rockylinux-x86_64"  
  sku       = "9-lvm"  
  version   = "9.3.20231113"  
}
```

## ➤ 이미지 생성오류

WEB1 을 통해 이미지를 등록해도 정상적으로 VM 이 일반화 되지 않았습니다. 이미지 일반화는 순차처리가 요구되는데 테라폼에서는 모듈 실행 순서가 지켜지지 않아 발생하는 문제였습니다. 따라서 depends\_on 코드로 순서에 맞게 실행되도록 하여 VMSS 이미지를 정상적으로 생성하는 것을 확인할 수 있었습니다.



depends\_on으로 순서 지정

그림 14 이미지 일반화의 순차처리 설계

## ➤ 유저데이터 복사오류

VM 생성 후 일반화에 즉시 돌입하여 발생하는 오류였습니다. 따라서 time\_sleep 모듈로 대기시간을 2 분 부여하여 정상적으로 VM 이 생성된 후에 일반화가 시작되도록 하였습니다.

```

| . 0 0.+0++|
| 0 + 00+0=|
| 0 *.+*=0|
| 0 0=+*+*|
+----[SHA256]-----+
Cloud-init v. 23.1.1-11.el9.0.1 running 'modules:config' at Sun, 23 Mar 2025 13:58:32 +0000. Up 19.83 seconds.
Cloud-init v. 23.1.1-11.el9.0.1 running 'modules:final' at Sun, 23 Mar 2025 13:58:34 +0000. Up 21.67 seconds.
+ setenforce 0
+ grubby --update-kernel ALL --args selinux=0
+ dnf config-manager --set-enabled crb
+ dnf install -y WALinuxAgent httpd wget tar php php-cli php-pdo php-fpm php-json php-mysqlnd
Rocky Linux 9 - BaseOS 716 kB/s | 2.3 MB 00:03
Rocky Linux 9 - AppStream 1.8 MB/s | 8.6 MB 00:04
Cloud-init v. 23.1.1-11.el9.0.1 running 'init-local' at Sun, 23 Mar 2025 14:01:07 +0000. Up 12.83 seconds.
Cloud-init v. 23.1.1-11.el9.0.1 running 'init' at Sun, 23 Mar 2025 14:01:10 +0000. Up 15.89 seconds.
ci-info: +-----+Net device info+-----+
ci-info: +-----+
ci-info: | Device | Up | Address | Mask | Scope | Hw-Address |
ci-info: +-----+
ci-info: | eth0 | True | 10.0.3.4 | 255.255.255.0 | global | 60:45:bd:44:fb:4e |
ci-info: | eth0 | True | fe80::6245:bdff:fe44:fb4e/64 | . | link | 60:45:bd:44:fb:4e |
ci-info: | lo | True | 127.0.0.1 | 255.0.0.0 | host | . |
ci-info: | lo | True | ::1/128 | . | host | . |
ci-info: +-----+
ci-info: +-----+Route IPv4 info+-----+

```

그림 15 일반화 생성오류 메시지



✓ 해결방안을 적용한 10\_web1gen.tf

```
resource "time_sleep" "wait_before_stop" {  
    create_duration = "120s"  
    depends_on = [azurerm_linux_virtual_machine.team3_web1]  
}  
  
resource "null_resource" "stop_web1" {  
    provisioner "local-exec" {  
        command = "az vm stop --resource-group 02-team3-rg --name team3-web1"  
    }  
    depends_on = [time_sleep.wait_before_stop]  
}  
  
resource "null_resource" "deal_web1" {  
    provisioner "local-exec" {  
        command = "az vm deallocate --resource-group 02-team3-rg --name team3-web1"  
    }  
    depends_on = [null_resource.stop_web1]  
}  
  
resource "null_resource" "gen_web1" {  
    provisioner "local-exec" {  
        command = "az vm generalize --resource-group 02-team3-rg --name team3-web1"  
    }  
    depends_on = [null_resource.deal_web1]  
}
```

## V. 참고문헌

---

1. Microsoft. (n.d.). *Microsoft Azure Portal*. Retrieved March 21, 2025  
<https://portal.azure.com>
2. Microsoft. (n.d.). *Design and implement Microsoft Azure networking solutions (AZ-700)*. Microsoft Learn. Retrieved March 21, 2025  
<https://learn.microsoft.com/ko-kr/training/paths/design-implement-microsoft-azure-networking-solutions-az-700/>
3. HashiCorp. (n.d.). *Azure Resource Manager (azurerm) provider*. Terraform Registry. Retrieved March 21, 2025  
<https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>
4. Microsoft. (n.d.). *Install the Azure CLI on Windows using winget*. Microsoft Learn. Retrieved March 21, 2025  
<https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows?pivots=winget>
5. Stack Overflow user. (2023, August 30). *Create a Rocky Linux virtual machine on Azure with Terraform?* Stack Overflow. Retrieved March 21, 2025  
<https://stackoverflow.com/questions/77016500/create-a-rocky-linux-virtual-machine-on-azure-with-terraform/78705990#78705990>