

Azure Terraform 실습

David Yoon | CEO
david@2miles.co.kr

실습 참고 자료

<https://bit.ly/3rvCxJ4>

<Ubuntu 설치>

Ubuntu 18.04 LTS / 크기는 B2s 2core 4GB 메모리 / 지역은 한국 중부

로그인 후 Azure CLI 설치

<https://docs.microsoft.com/ko-kr/cli/azure/install-azure-cli-apt?view=azure-cli-latest>

```
apt-get update -y
```

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

```
wget https://releases.hashicorp.com/terraform/0.13.3/terraform\_0.13.3\_linux\_amd64.zip
```

```
apt install unzip
```

```
unzip terraform_0.13.3_linux_amd64.zip
```

```
export PATH=$PATH:~/
```

홈 >

새로 만들기

Marketplace 검색

Azure Marketplace 모두 보기 인기

시작

최근에 만들어짐

AI + 기계 학습

분석

블록체인

컴퓨팅

컨테이너

데이터베이스

개발자 도구

DevOps

ID

통합

사물 인터넷

관리 도구

미디어

Migration

Mixed Reality

Monitoring & Diagnostics

네트워킹

보안



Windows Server 2016 Datacenter

빠른 시작 + 자습서



Ubuntu Server 18.04 LTS

자세한 정보



웹 앱

빠른 시작 + 자습서



SQL Database

빠른 시작 + 자습서



함수 앱

빠른 시작 + 자습서



Azure Cosmos DB

빠른 시작 + 자습서



Kubernetes Service

빠른 시작 + 자습서



DevOps Starter

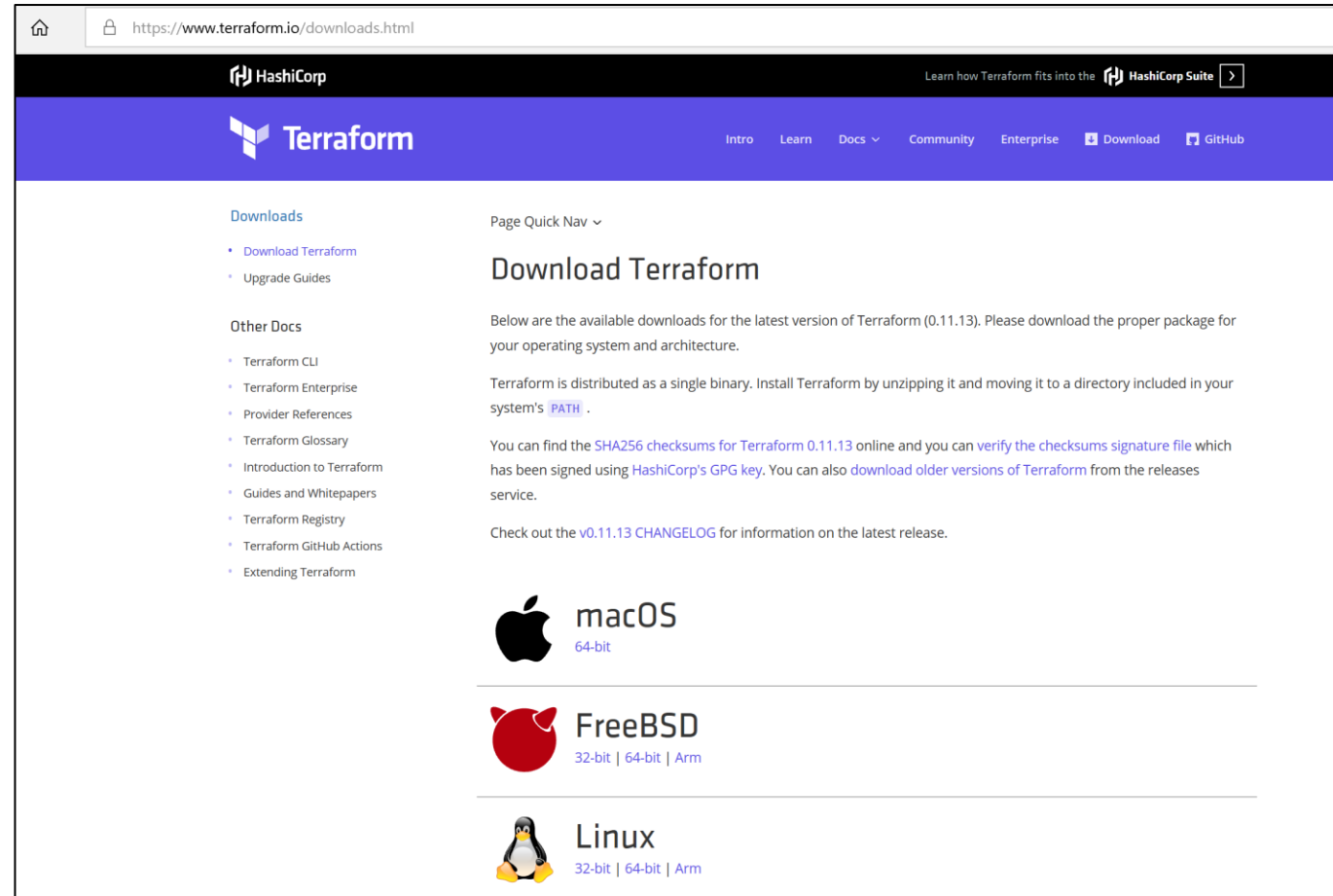
빠른 시작 + 자습서

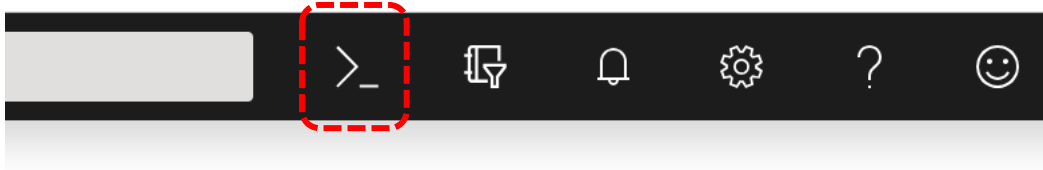


스토리지 계정

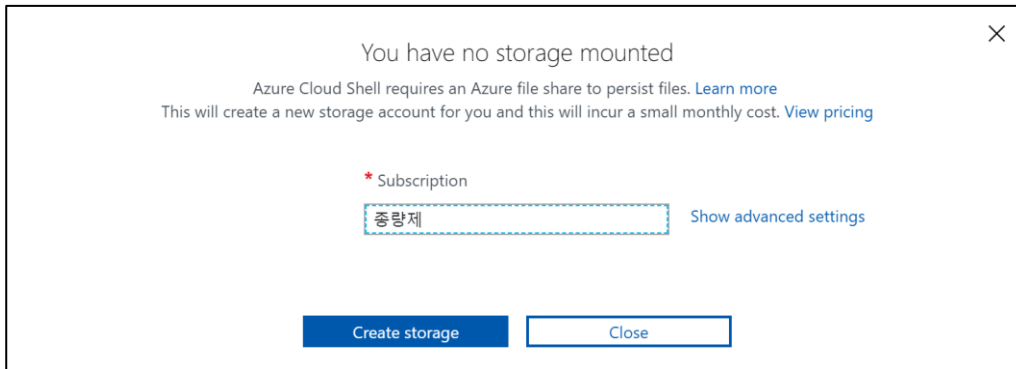
빠른 시작 + 자습서

- Terraform은 Single 파일로 동작
- Terraform 홈페이지에서 다운로드





- Azure Portal에서 ">_" 클릭



- Azure 구독 선택 후 Azure Cloud Shell에서 생성되는 데이터를 저장할 저장소 생성

```
Bash
Your cloud drive has been created in:

Subscription Id: e0fae348-f6c2-45f5-87b7-c41c22782d8f
Resource group: SKCNC
Storage account: skcncterraform
File share: skcncterraform

Initializing your account for Cloud Shell...\
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI 2.0
Type "help" to learn about Cloud Shell

user01@Azure:~$
```

Terraform에서 Azure에 리소스를 생성하기 위해서는 인증 정보가 필요

- Subscription ID
- Tenant ID
- Client ID
- Client Secret

Azure Subscription, Tenant, Client ID 설정

Azure CLI를 설치 후 아래 "az login" 명령어로 CLI에서 Azure 로그인

\$ az login (PC나 노트북에서 Azure CLI 설치하신 경우 Azure Login 필요, Azure Cloudshell은 이미 로그인 상태라 불필요)

위 명령어를 실행하면 웹브라우저에서 <https://Microsoft.com/devicelogin> 에서 인증 코드를 넣으라고 나옴
아래 명령줄에서 HJQT8L68E가 바로 인증 코드

인증코드를 넣으면, 로그인이 되고 Subscription ID, Tenant ID를 확인 할 수 있음

\$ az account list

```
gkn@Azure:~$ az login
Cloud Shell is automatically authenticated under the initial account signed-in with. Run 'az login' only if you need to use a different account
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code HJQT8L68E to authenticate.
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "029675e3-1fa6-457d-a5cd-bdabb958e953",
    "id": "1c2a5318-1[REDACTED]-18882c6ddbe9",
    "isDefault": true,
    "managedByTenants": [],
    "name": "종량제 1",
    "state": "Enabled",
    "tenantId": "029675e3-1[REDACTED]-bdabb958e953",
    "user": {
      "name": "gkn2025@hotmail.com",
      "type": "user"
    }
  }
]
```


Azure Subscription, Tenant, Client ID 설정

복수의 구독을 사용하는 경우, 사용할 구독 지정 필요

```
$ az account set --subscription="SUBSCRIPTION_ID"
```

```
gkn@Azure:~$ az account set --subscription="1c2a5318-██████████-18882c6ddbe9"
```

Azure AD Service Principal 생성 (리소스 관리 권한 할당) 을 통해 Client ID 및 Client Secret 생성

```
$ az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/SUBSCRIPTION_ID"
```

appId = ClientID, password = Client Secret

```
gkn@Azure:~$ az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/1c2a5318-fcca-42b7-a64e-18882c6ddbe9"
Creating a role assignment under the scope of "/subscriptions/1c2a5318-fcca-42b7-a64e-18882c6ddbe9"
Retrying role assignment creation: 1/36
{
  "appId": "7fb846e3-6e71-4d5e-██████████722bcab",
  "displayName": "azure-cli-2020-08-29-12-03-17",
  "name": "http://azure-cli-2020-08-29-12-03-17",
  "password": "G-87-ZrZosELziZ██████████uuHUIg~v",
  "tenant": "029675e3-1fa6-457d-a5cd-bdabb958e953"
}
```

Azure Subscription, Tenant, Client ID 설정

- Subscription(구독) 탭에서 구독 ID 확인

Microsoft Azure

Search resources, services, and docs

Home > Subscriptions

Subscriptions

기본 디렉터리

+ Add

Showing subscriptions in 기본 디렉터리. Don't see a subscription? [Switch directories](#)

My role ⓘ

8 selected

Apply

☒ Show only subscriptions selected in the [global subscriptions filter](#) ⓘ

Search to filter items...

SUBSCRIPTION	↑↓	SUBSCRIPTION ID
종량제		1c2a5318-fcca-[REDACTED]-18882c6ddbe9

`provider` 섹션은 Azure 공급자를 사용하도록 Terraform에 전달
`subscription_id`, `client_id`, `client_secret` 및 `tenant_id`에 대한 값 설정

main.tf. (주의! github public repository에 절대 올리시지 마세요!!!)

```
provider "azurerm" {  
  subscription_id = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"  
  client_id       = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"  
  client_secret   = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"  
  tenant_id       = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"  
  features {}  
}
```

Azure 연결 방법2 – provider.tf (아래 방법 추천)

`provider` 섹션은 Azure 공급자를 사용하도록 Terraform에 전달
`subscription_id`, `client_id`, `client_secret` 및 `tenant_id`에 대한 값 설정

main.tf.

```
provider "azurerm" {  
    features {}  
}
```

CLI에서 전역 변수로 설정

```
export ARM_CLIENT_ID="00000000-0000-0000-0000-000000000000"  
export ARM_CLIENT_SECRET="00000000-0000-0000-0000-000000000000"  
export ARM_SUBSCRIPTION_ID="00000000-0000-0000-0000-000000000000"  
export ARM_TENANT_ID="00000000-0000-0000-0000-000000000000"
```

Region 설정 후 리소스 그룹 이름을 지정하여 아래와 같이 생성

```
$terraform init
```

resource.tf

```
resource "azurerm_resource_group" "user01-rg" {  
  name      = "user01resourcegroup"  
  location  = "koreacentral"  
  
  tags = {  
    environment = "Terraform Demo"  
  }  
}
```

Azure Region 코드는 아래 명령어로 확인 가능

```
$az account list-locations
```

```
user01@Azure:~$ mkdir azure
user01@Azure:~$ cd azure/
user01@Azure:~/azure$ mkdir demo01
user01@Azure:~/azure$ cd demo01/
user01@Azure:~/azure/demo01$ vi main.tf
user01@Azure:~/azure/demo01$ terraform init
```

Initializing provider plugins...

- Checking for available provider plugins on <https://releases.hashicorp.com...>
- Downloading plugin for provider "azurerm" (1.24.0)...

The following providers do not have any version constraints in configuration, so the latest version was installed.

```
user01@Azure:~/azure/demo01$ vi resource.tf
user01@Azure:~/azure/demo01$ cat resource.tf
resource "azurerm_resource_group" "user01-rg" {
  name      = "user01resourcegroup"
  location  = "koreacentral"

  tags {
    environment = "Terraform Demo"
  }
}
user01@Azure:~/azure/demo01$ terraform init
```

```
user01@Azure:~/azure/demo01$ terraform apply
```

```
An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:
```

```
+ create
```

```
Terraform will perform the following actions:
```

```
+ azurerm_resource_group.myterraformgroup
```

```
id: <computed>
```

```
location: "koreacentral"
```

```
name: "user01resourcegroup"
```

```
tags.%: "1"
```

```
tags.environment: "Terraform Demo"
```

```
Plan: 1 to add, 0 to change, 0 to destroy.
```

```
Do you want to perform these actions?
```

```
Terraform will perform the actions described above.
```

```
Only 'yes' will be accepted to approve.
```

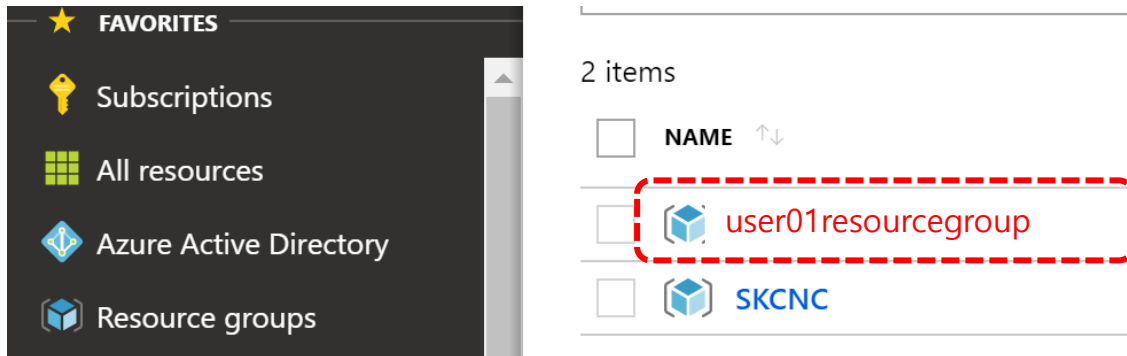
```
Enter a value: yes
```

- Terraform 실행을 위해 "yes" 입력

```
azurerm_resource_group.user01-rg: Creating...
  location:      "" => "koreacentral"
  name:          "" => "user01resourcegroup"
  tags.%:        "" => "1"
  tags.environment: "" => "Terraform Demo"
azurerm_resource_group.myterraformgroup: Creation complete after 3s (1s)

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
user01@Azure:~/azure/demo01$
```

- Azure Portal에서 리소스 그룹 생성 확인



myVnet.tf

```
resource "azurerm_virtual_network" "user01-vnet" {  
    name                = "user01-myVnet"  
    address_space       = ["1.0.0.0/16"]  
    location             = azurerm_resource_group.user01-rg.location  
    resource_group_name = azurerm_resource_group.user01-rg.name  
}
```

- terraform plan

```
user01@Azure:~/azure/demo01$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan,
persisted to local or remote state storage.

azurerm_resource_group.myterraformgroup: Refreshing state
-----

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

+ azurerm_virtual_network.myterraformnetwork
  id: <computed>
  address_space.#: "1"
  address_space.0: "1.0.0.0/16"
  location: "koreacentral"
  name: "myVnet"
  resource_group_name: "user01resourcegroup"
  subnet.#: <computed>
  tags.%: "1"
  tags.environment: "Terraform Demo"

Plan: 1 to add, 0 to change, 0 to destroy.
```

- terraform apply

```
user01@Azure:~/azure/demo01$ terraform apply
azurerm_resource_group.myterraformgroup: Refreshing state...

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

+ azurerm_virtual_network.myterraformnetwork
  id: <computed>
  address_space.#: "1"
  address_space.0: "1.0.0.0/16"
  location: "koreacentral"
  name: "myVnet"
  resource_group_name: "user01resourcegroup"
  subnet.#: <computed>
  tags.%: "1"
  tags.environment: "Terraform Demo"

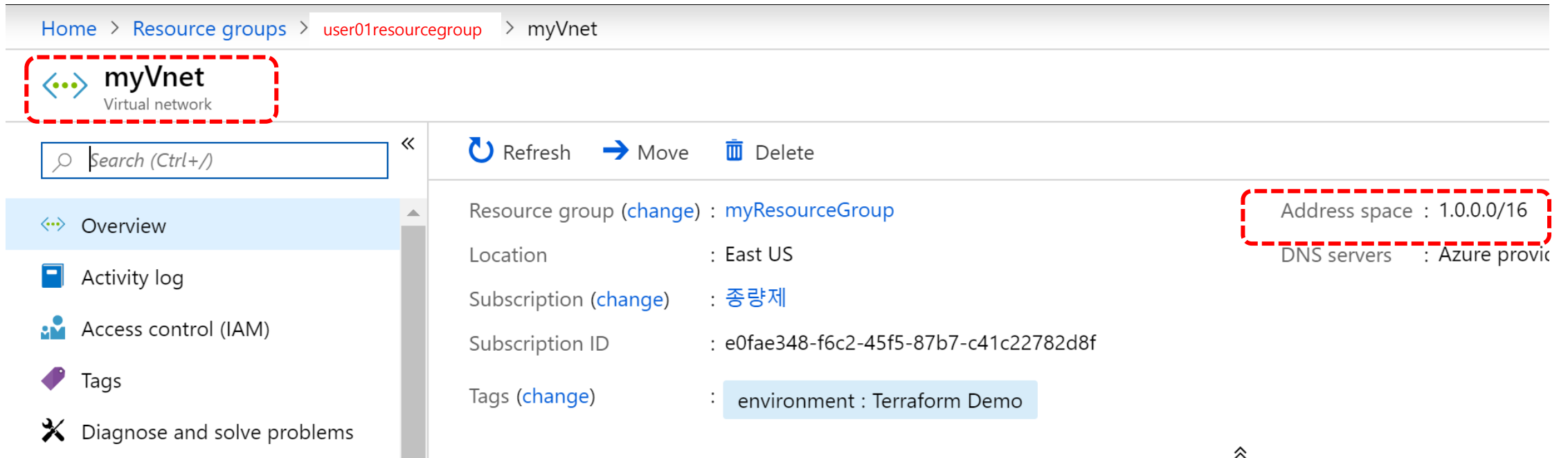
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_virtual_network.myterraformnetwork: Creating...
  address_space.#: "" => "1"
  address_space.0: "" => "1.0.0.0/16"
```

- Azure Portal에서 가상 네트워크 생성 확인



Home > Resource groups > user01resourcegroup > myVnet

myVnet
Virtual network

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Refresh Move Delete

Resource group (change) : myResourceGroup

Location : East US

Subscription (change) : 종량제

Subscription ID : e0fae348-f6c2-45f5-87b7-c41c22782d8f

Tags (change) : environment : Terraform Demo

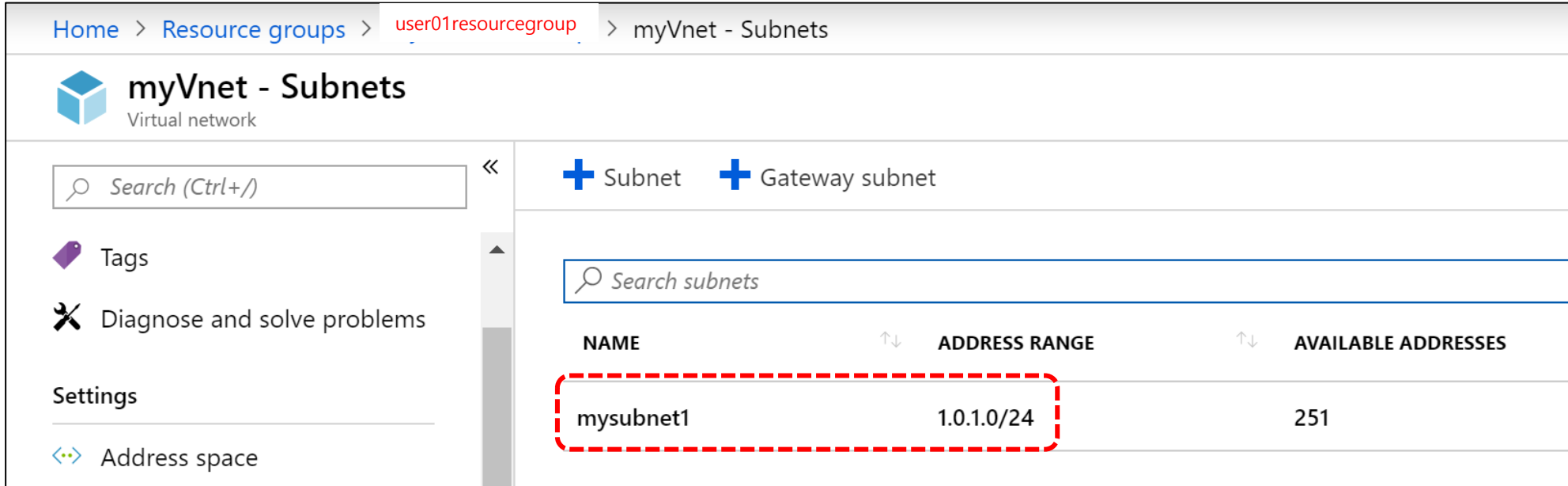
Address space : 1.0.0.0/16

DNS servers : Azure provid

mySubnet.tf (예, subnet은 *.0.1.0/24로 설정 => user01은 1.0.1.0/24)

```
resource "azurerm_subnet" "user01-subnet1" {  
  name                       = "user01-mysubnet1"  
  resource_group_name        = azurerm_resource_group.user01-rg.name  
  virtual_network_name       = azurerm_virtual_network.user01-vnet.name  
  address_prefixes            = ["1.0.1.0/24"]  
}
```

- Azure Portal에서 서브넷 생성 확인



Home > Resource groups > user01resourcegroup > myVnet - Subnets

myVnet - Subnets
Virtual network

Search (Ctrl+ /)

Tags

Diagnose and solve problems

Settings

Address space

+ Subnet + Gateway subnet

Search subnets

NAME	ADDRESS RANGE	AVAILABLE ADDRESSES
mysubnet1	1.0.1.0/24	251

public_ip.tf

```
resource "azurerm_public_ip" "user01-publicip" {  
  name          = "mypublicIP"  
  location      = azurerm_resource_group.user01-rg.location  
  resource_group_name = azurerm_resource_group.user01-rg.name  
  allocation_method = "Static"  
  domain_name_label = azurerm_resource_group.user01-rg.name  
  
  ## 동일 Region에 추가 Public IP 생성시에는 아래 내용 수정 (위 설정했으면 아래는 #처리)  
  #domain_name_label = "user01pubip2.westus.cloudapp.azure.com"  
  
  tags = {  
    environment = "staging"  
  }  
}
```

myNetworkSecurityGroup.tf

```
resource "azurerm_network_security_group" "user01nsg" {
  name                = "user01nsg"
  location            = azurerm_resource_group.user01-rg.location
  resource_group_name = azurerm_resource_group.user01-rg.name

  security_rule {
    name                = "SSH"
    priority            = 1001
    direction          = "Inbound"
    access             = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "22"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }
  security_rule {
    name                = "HTTP"
    priority            = 2001
    direction          = "Inbound"
    access             = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "80"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }

  tags = {
    environment = "Terraform Demo"
  }
}
```

lb.tf

```
resource "azurerm_lb" "user01-lb" {  
  name                = "user01lb"  
  location            = azurerm_resource_group.user01-rg.location  
  resource_group_name = azurerm_resource_group.user01-rg.name  
  
  frontend_ip_configuration {  
    name                = "user01PublicIPAddress"  
    public_ip_address_id = azurerm_public_ip.user01-publicip.id  
  }  
}
```

lb_backendpool.tf

```
resource "azurerm_lb_backend_address_pool" "user01-bpepool" {  
  resource_group_name = azurerm_resource_group.user01-rg.name  
  loadbalancer_id = azurerm_lb.user01-lb.id  
  name = "user01-BackEndAddressPool"  
}
```



백 엔드 풀	가상 머신	가상 머신 상태	네트워크 인터페이스	프라이빗 IP 주소
▼ user03-BackEndAddressPool(2대 가상 ...				
	mytestscalest-1(인스턴스 0)	실행 중	terraformnetworkprofile	3.0.1.4
	mytestscalest-1(인스턴스 2)	실행 중	terraformnetworkprofile	3.0.1.6

lb_natpool.tf

```
resource "azurerm_lb_nat_pool" "lbnatpool" {  
  resource_group_name = azurerm_resource_group.user01-rg.name  
  name = "ssh"  
  loadbalancer_id = azurerm_lb.user01-lb.id  
  protocol = "Tcp"  
  frontend_port_start = 50000  
  frontend_port_end = 50119  
  backend_port = 22  
  frontend_ip_configuration_name = "user01PublicIPAddress"  
}
```

+ 추가 새로 고침

이름으로 필터링...

이름	IP 버전	대상 주소	대상 컴퓨터	서비스
ssh.0	IPv4	20.194.58.168	mytestscaleset-1(인스턴스 0)	사용자 지정(TCP/50000)
ssh.2	IPv4	20.194.58.168	mytestscaleset-1(인스턴스 2)	사용자 지정(TCP/50002)

lb_probe.tf

```
resource "azurerm_lb_probe" "user01-lb-probe" {  
  resource_group_name = azurerm_resource_group.user01-rg.name  
  loadbalancer_id = azurerm_lb.user01-lb.id  
  name = "http-probe"  
  protocol = "Http"  
  request_path = "/"  
  port = 80  
}
```

http-probe

user03lb

저장 취소 삭제

이름 *

http-probe

프로토콜 ⓘ

HTTP

포트 * ⓘ

80

경로 * ⓘ

/

간격 * ⓘ

15

초

비정상 임계값 * ⓘ

2

연속 오류

사용자 ⓘ

사용 안 됨

lb_rule.tf

```
resource "azurerm_lb_rule" "lbnatrue" {  
  resource_group_name = azurerm_resource_group.user01-rg.name  
  loadbalancer_id = azurerm_lb.user01-lb.id  
  name = "http"  
  protocol = "Tcp"  
  frontend_port = 80  
  backend_port = 80  
  backend_address_pool_id = azurerm_lb_backend_address_pool.user01-bpepool.id  
  frontend_ip_configuration_name = "user01PublicIPAddress"  
  probe_id = azurerm_lb_probe.user01-lb-probe.id  
}
```

web.sh (VMSS 만들기 전에 미리 생성, 실제 가상서버에서 아래 명령어가 실행됩니다)

```
#!/bin/bash
```

(Ubuntu 혹은 CentOS 한가지만 선택, 나머지는 주석처리)

#Ubuntu 사용하시는 경우

```
apt-get update -y
```

```
apt-get install -y apache2
```

#CentOS, RHEL 사용하시는 경우

```
#yum update -y
```

```
#yum install -y apache2
```

```
echo "<html>" > /var/www/html/index.html ## 리다이렉션 반드시 한개로
```

```
echo "Hello World from $(hostname -f)" >> /var/www/html/index.html
```

```
echo "</html>" >> /var/www/html/index.html
```

*** Cloudshell에서 테스트로 실행시 권한이 없어서 에러가 발생합니다.**

vmss.tf

```
resource "azurerm_virtual_machine_scale_set" "user01vmss" {  
  name = "user01vmss"  
  location = azurerm_resource_group.user01-rg.location  
  resource_group_name = azurerm_resource_group.user01-rg.name  
  upgrade_policy_mode = "Manual"  
}
```

계속 진행



vmss.tf

```
sku {  
  name = "Standard_D2_v3"  
  tier = "Standard"  
  capacity = 2  
}  
storage_profile_image_reference {  
  publisher = "Canonical"  
  offer = "UbuntuServer"  
  sku = "18.04-LTS"  
  version = "latest"  
}
```

계속 진행



vmss.tf

```
storage_profile_os_disk {  
  name = ""  
  caching = "ReadWrite"  
  create_option = "FromImage"  
  managed_disk_type = "Standard_LRS"  
}  
storage_profile_data_disk {  
  lun = 0  
  caching = "ReadWrite"  
  create_option = "Empty"  
  disk_size_gb = 10  
}  
os_profile {  
  computer_name_prefix = "testvm"  
  admin_username = "myadmin" ## VM 에 접속할 계정  
  custom_data = file("web.sh")  
}
```

계속 진행



vmss.tf

```
os_profile_linux_config {
  disable_password_authentication = true
  ssh_keys {
    path = "/home/myadmin/.ssh/authorized_keys"  ## pwd 실행후 경로설정 ex) /home/user01 등
    key_data = file("~/.ssh/id_rsa.pub")  ## 터미널에서 ssh-keygen 으로 생성 (엔터 3번)
  }
}
network_profile {
  name = "terraformnetworkprofile"
  primary = true
  ip_configuration {
    name = "TestIPConfiguration"
    primary = true
    subnet_id = azurerm_subnet.user01-subnet1.id
    load_balancer_backend_address_pool_ids = [azurerm_lb_backend_address_pool.user01-bpepool.id]
    load_balancer_inbound_nat_rules_ids = [azurerm_lb_nat_pool.lbnatpool.id]
  }
  network_security_group_id = azurerm_network_security_group.user01nsg.id
}
tags = {
  environment = "staging"
}
```


참고자료

(지금 부터는 참고자료)

myPublicIP.tf

```
resource "azurerm_public_ip" "example" {  
  name = "myPublicIP"  
  location = azurerm_resource_group.example.location  
  resource_group_name = azurerm_resource_group.example.name  
  allocation_method = "Static"  
  domain_name_label = azurerm_resource_group.example.name  
  
  tags = {  
    environment = "staging"  
  }  
}
```

- Azure Portal에서 Public IP 생성 확인

<input type="checkbox"/>	NAME <small>↑↓</small>	TYPE <small>↑↓</small>	LOCATION <small>↑↓</small>
<input type="checkbox"/>	 myPublicIP	Public IP address	East US
<input type="checkbox"/>	 myVnet	Virtual network	East US

myNetworkSecurityGroup.tf


```
resource "azurerm_network_security_group" "myterraformnsg" {
  name                = "myNetworkSecurityGroup"
  location            = "koreacentral"
  resource_group_name = azurerm_resource_group.user01-rg.name

  security_rule {
    name                = "SSH"
    priority            = 1001
    direction          = "Inbound"
    access              = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "22"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }

  tags = {
    environment = "Terraform Demo"
  }
}
```

- Azure Portal에서 Network Security Group 생성 확인

[Home](#) > [Resource groups](#) > [myResourceGroup](#) > myNetworkSecurityGroup

 **myNetworkSecurityGroup**
Network security group

Search (Ctrl+[/](#))

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Inbound security rules






Outbound security rules

[Move](#) [Delete](#) [Refresh](#)

Tags ([change](#)) : environment : Terraform Demo

⌵

Inbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
1001	 SSH	22	TCP	Any	Any	 Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	 Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	 Allow
65500	DenyAllInBound	Any	Any	Any	Any	 Deny

myNetworkSecurityGroup.tf


```
resource "azurerm_network_security_group" "user01nsg" {
  name           = "user01nsg"
  location       = "koreacentral"
  resource_group_name = azurerm_resource_group.user01-rg.name

  security_rule {
    name           = "SSH"
    priority       = 1001
    direction     = "Inbound"
    access         = "Allow"
    protocol       = "Tcp"
    source_port_range = "*"
    destination_port_range = "22"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }
  security_rule {
    name           = "HTTP"
    priority       = 2001
    direction     = "Inbound"
    access         = "Allow"
    protocol       = "Tcp"
    source_port_range = "*"
    destination_port_range = "80"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }

  tags = {
    environment = "Terraform Demo"
  }
}
```

Azure 보안 그룹 생성 – Rule 추가

[Home](#) > [Resource groups](#) > [myResourceGroup](#) > myNetworkSecurityGroup

 **myNetworkSecurityGroup**
Network security group

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Inbound security rules

Outbound security rules

Network interfaces

Subnets

Properties

Locks

Export template

[Move](#) [Delete](#) [Refresh](#)

Resource group ([change](#)) : myResourceGroup

Location : East US

Subscription ([change](#)) : 종량제

Subscription ID : e0fae348-f6c2-45f5-87b7-c41c22782d8f

Tags ([change](#)) : environment : Terraform Demo

Custom security rules : 2 inbound, 0 outbound

Associated with : 0 subnets, 0 network interfaces

Inbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
1001	SSH	22	TCP	Any	Any	Allow
2001	HTTP	80	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

myNIC.tf

```
resource "azurerm_network_interface" "myterraformnic" {
  name            = "myNIC"
  location        = "eastus"
  resource_group_name = "${azurerm_resource_group.myterraformgroup.name}"

  ip_configuration {
    name                = "myNicConfiguration"
    subnet_id           = "${azurerm_subnet.myterraformsubnet.id}"
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id = "${azurerm_public_ip.myterraformpublicip.id}"
  }

  tags = {
    environment = "Terraform Demo"
  }
}
```

```
resource "azurerm_network_interface_security_group_association" "example" {
  network_interface_id      = azurerm_network_interface.myterraformnic.id
  network_security_group_id = azurerm_network_security_group.myterraformnsg.id
}
```

- Azure Portal에서 NIC생성 확인

Home > Resource groups > myResourceGroup > myNIC > myPublicIP > myNIC - IP configurations

myNIC - IP configurations

Network interface

Search (Ctrl+ /) << + Add Save X Discard

Overview

Activity log

Access control (IAM)

Tags

Settings

IP configurations

DNS servers

Network security group

Properties

IP forwarding settings

IP forwarding Disabled Enabled

Virtual network [myVnet](#)

IP configurations

* Subnet

Search IP configurations

NAME	IP VERSION	TYPE	PRIVATE IP ADDRESS	PUBLIC IP ADDRESS
myNicConfigur...	IPv4	Primary	1.0.1.4 (Dynamic)	13.68.220.246 (myPublicIP)

VM을 위한 부트 진단을 저장하려면 저장소 계정이 필요합니다. 이러한 부트 진단은 문제를 해결하고 VM의 상태를 모니터링하는 데 도움을 줄 수 있습니다. 사용자가 만든 저장소 계정은 부팅 진단 데이터를 저장하기 위한 것입니다. 각 저장소 계정에는 고유한 이름을 부여하기 위해 random ID를 생성합니다.

myRandomID.tf


```
resource "random_id" "randomId" {
  keepers = {
    # Generate a new ID only when a new resource group is defined
    resource_group = "${azurerm_resource_group.myterraformgroup.name}"
  }

  byte_length = 8
}
```

myDiagStorage.tf

```
resource "azurerm_storage_account" "mystorageaccount" {  
  name                = "diag${random_id.randomId.hex}"  
  resource_group_name = "${azurerm_resource_group.myterraformgroup.name}"  
  location             = "eastus"  
  account_replication_type = "LRS"  
  account_tier         = "Standard"  
  
  tags {  
    environment = "Terraform Demo"  
  }  
}
```

- Azure Portal에서 diag 저장소 확인

<input type="checkbox"/>	NAME <small>↑↓</small>	TYPE <small>↑↓</small>	LOCATION <small>1</small>
<input type="checkbox"/>	 diagb79d22d11e440b69	Storage account	East US

```
resource "azurerm_virtual_machine" "myterraformvm" {
  name                = "myVM"
  location            = "eastus"
  resource_group_name = "${azurerm_resource_group.myterraformgroup.name}"
  network_interface_ids = ["${azurerm_network_interface.myterraformnic.id}"]
  vm_size             = "Standard_DS1_v2"

  storage_os_disk {
    name          = "myOsDisk"
    caching       = "ReadWrite"
    create_option = "FromImage"
    managed_disk_type = "Premium_LRS"
  }
  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04.0-LTS"
    version   = "latest"
  }
}
```

Azure 가상서버 만들기 - myVM.tf (2/2)

```
os_profile {
  computer_name = "myvm"
  admin_username = "azureuser"
  admin_password = "*****"
}

os_profile_linux_config {
  disable_password_authentication = true
  ssh_keys {
    path = "/home/azureuser/.ssh/authorized_keys"
    key_data = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBAQCgkNw/pUD8ld/cJflpD/A1yZSsONcfDjhW4uS27QSbfj11J/oVR21VzdEcqeOYQSgLngVxVRhLEftRslZM2K
8EU1M8uRX1hte/tBM/jLK5J4n35tFiQvhzevCZUcygx2F4BAXGv1z0vebL8s7oCeBhK8Ntv30wCDGquh4dRka1vgxj+OIEY7ELcGAUztZblwDe0GFdMh
HdESmLfuClo2waRsEYr63PnCmVwEZLdBk6Cjhk3RUhV4fWiVS1pvINGMxpTQA7N7dX/8IW2gjRx+u3QQ0053fKBh4HcjQyU2bCGEU9VAj3iyYI2+vM
0O7L0nB8qV95a3ew/UDCV8UOYFkR user01@cc-493c9167-8558cffbf7-v9czj"
  }
}

boot_diagnostics {
  enabled = "true"
  storage_uri = "${azurerm_storage_account.mystorageaccount.primary_blob_endpoint}"
}

tags {
  environment = "Terraform Demo"
}
```

참고> ssh-keygen 사용해서 RSA Priv/Pub Key 생성

Bash



```
user01@Azure:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user01/.ssh/id_rsa):
Created directory '/home/user01/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user01/.ssh/id_rsa.
Your public key has been saved in /home/user01/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:i6YbZ9lddjdYI2gAmT3UB+DLDDjyC0VDAGffvbIEfg0 user01@cc-493c9167-8558cffbf7-v9czj
The key's randomart image is:
+---[RSA 2048]---+
|  ..+O+  .*+O..  |
|    O O +OOO....  |
|     . * E O.O.. O  |
|    = O * +    + .  |
|     . O +S*   O....  |
|     . ++OO O . . .  |
|     ..*.O .        |
|      *             |
|     O.             |
+-----[SHA256]-----+
```

```
user01@Azure:~/azure/demo01$ ssh -i ~/.ssh/id_rsa azureuser@23.96.33.148
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1040-azure x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.
```

- GitHub VM 생성 하기 - <https://github.com/Azure/terraform-azurerm-vm>
- VM 따라 만들기 - <https://bit.ly/2X8nFAg>

감사합니다

