Groundbreakers

16th Oracle Developer Meetup

Infrastructure as Code

강인호
inho.kang@oracle.com

2020.01.18

ORACLE

Kang In Ho

- .Net Developer
- CBD, SOA Methodology Consulting
- ITA/EA, ISP Consulting
- Oracle Corp.
  - Middleware
  - Cloud Native Application, Container Native
  - Emerging Technology Team
- k8s korea user group

innoshom@gamil.com

A Table of Contents

PART 1

Infra as Code

오늘의 컨셉

오늘 다루지 않는 내용

챙겨줄게요
처음부터 끝까지

History of IaC

기원

용어 정의를 누가 했느냐는 명확하지 않지만 2009 Velocity Conference에서 나오게 되었고, Cloud 환경이 도입되면서, pace of change, automation의 요구

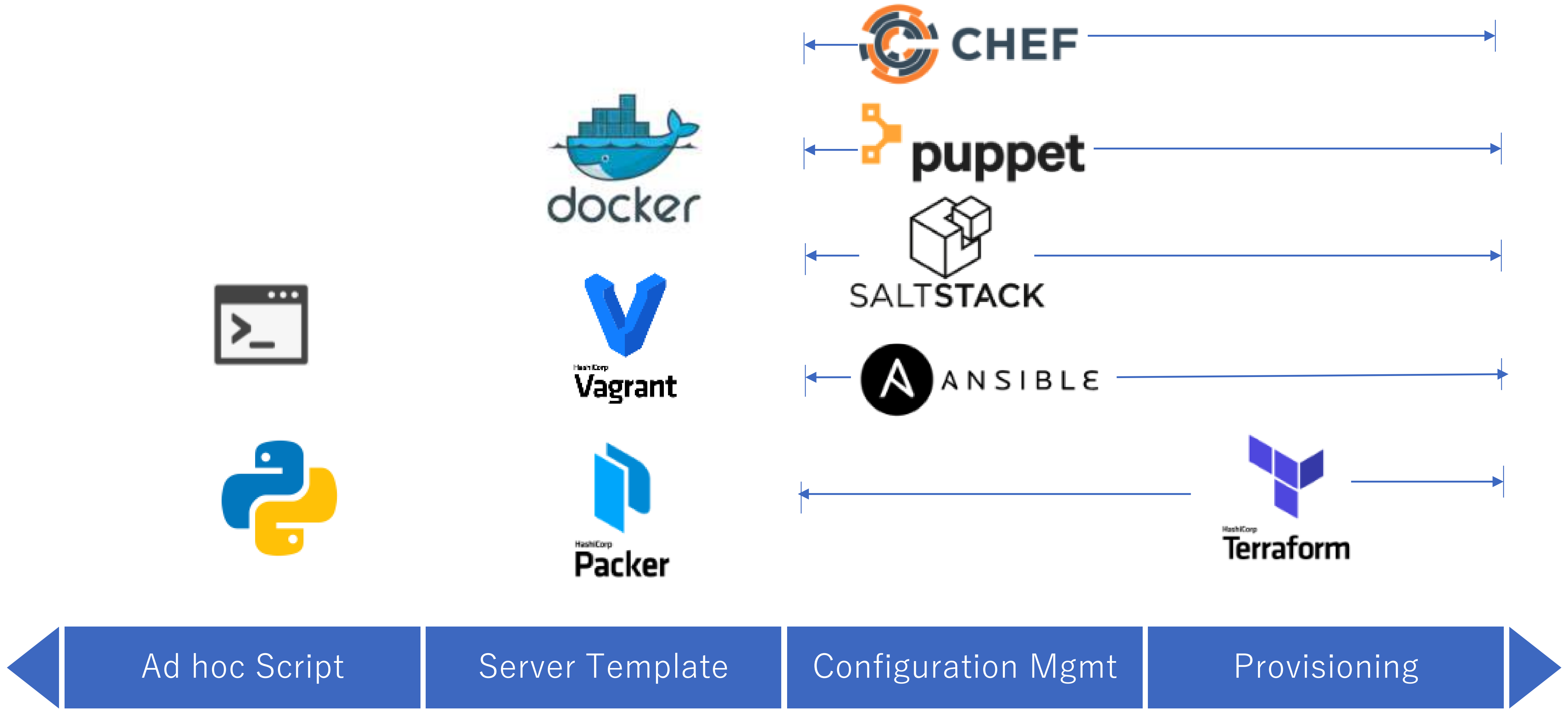
- from : Infrastructure as Code

Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

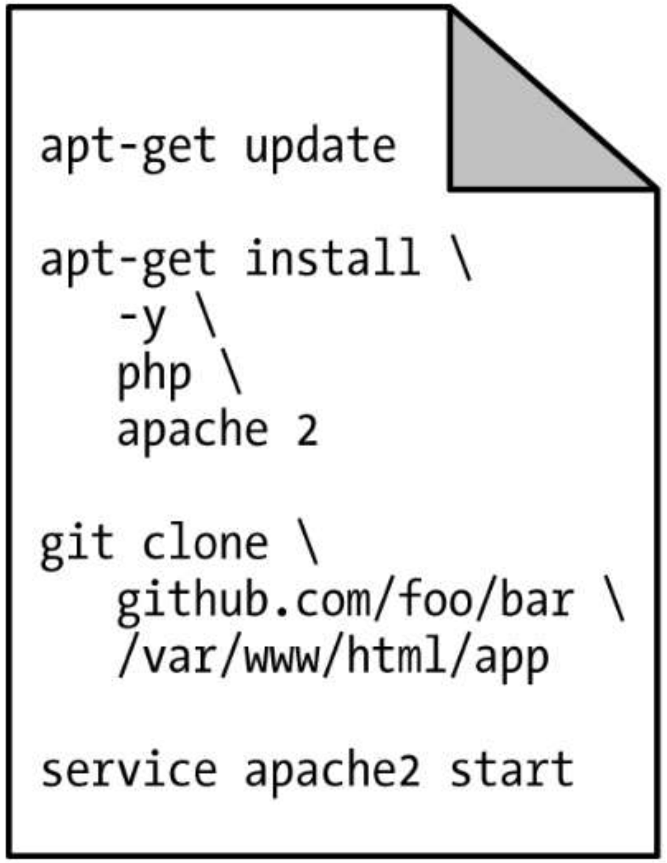
The IT infrastructure managed by this comprises both physical equipment such as bare-metal servers as well as virtual machines and associated configuration resources. The definitions may be in a version control system. It can use either scripts or declarative definitions, rather than manual processes, but the term is more often used to promote declarative approaches.
IaC approaches are promoted for cloud computing, which is sometimes marketed as infrastructure as a service (IaaS). IaC supports IaaS, but should not be confused with it.

- from : Wikipedia

IaC Tools

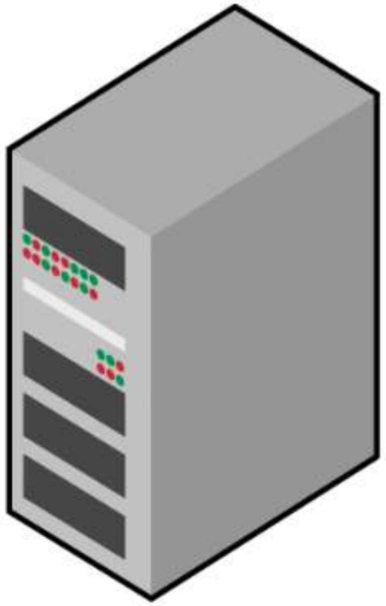

CHEF
puppet
docker
SALTSTACK
HashiCorp Vagrant
ANSIBLE
HashiCorp Packer
HashiCorp Terraform
Ad hoc Script
Server Template
Configuration Mgmt
Provisioning

IaC Tools – Ad hoc script



```
apt-get update

apt-get install \
    -y \
    php \
    apache 2

git clone \
    github.com/foo/bar \
    /var/www/html/app

service apache2 start
```

Ad hoc script

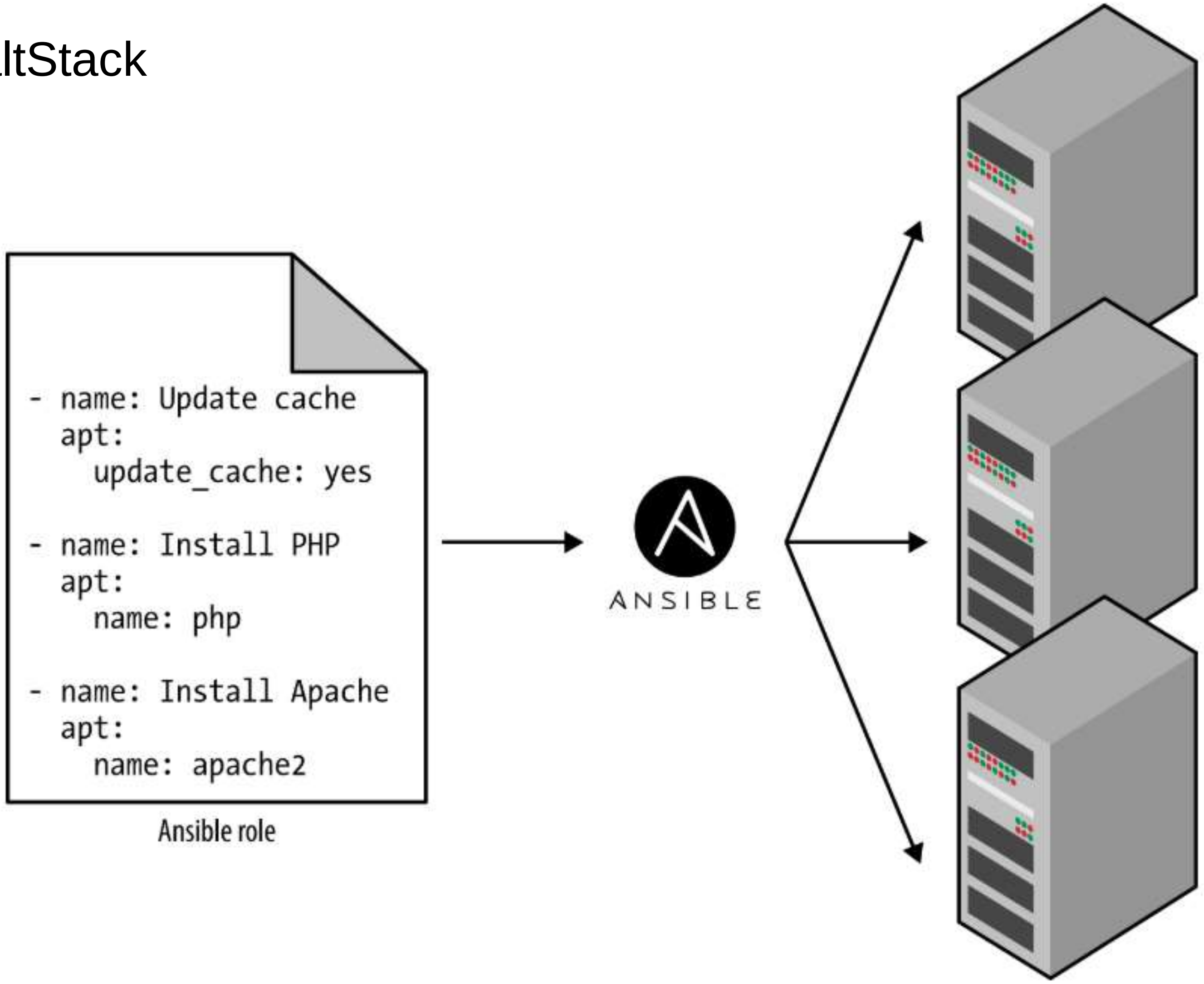


- 다양한 스크립트 언어 사용 (bash, python, ruby등)
- 모든 코드를 직접 작성
- 한두대는 문제가 없지만.....
  - 시스템이 커지만 스파게티 코드

IaC Tools – Configuration Management Tools

Chef, Puppet, Ansible, and SaltStack

- 추상화와 자동화를 위해
  DSL(Domain Specfic Language) 사용
- 멱등성(Idempotent)
  여러분 수행해도 같은 결과가 나온다
- 대규모 분산환경을 위해 설계
- 확장이 간편함
  서버 댓수 설정만으로 확장

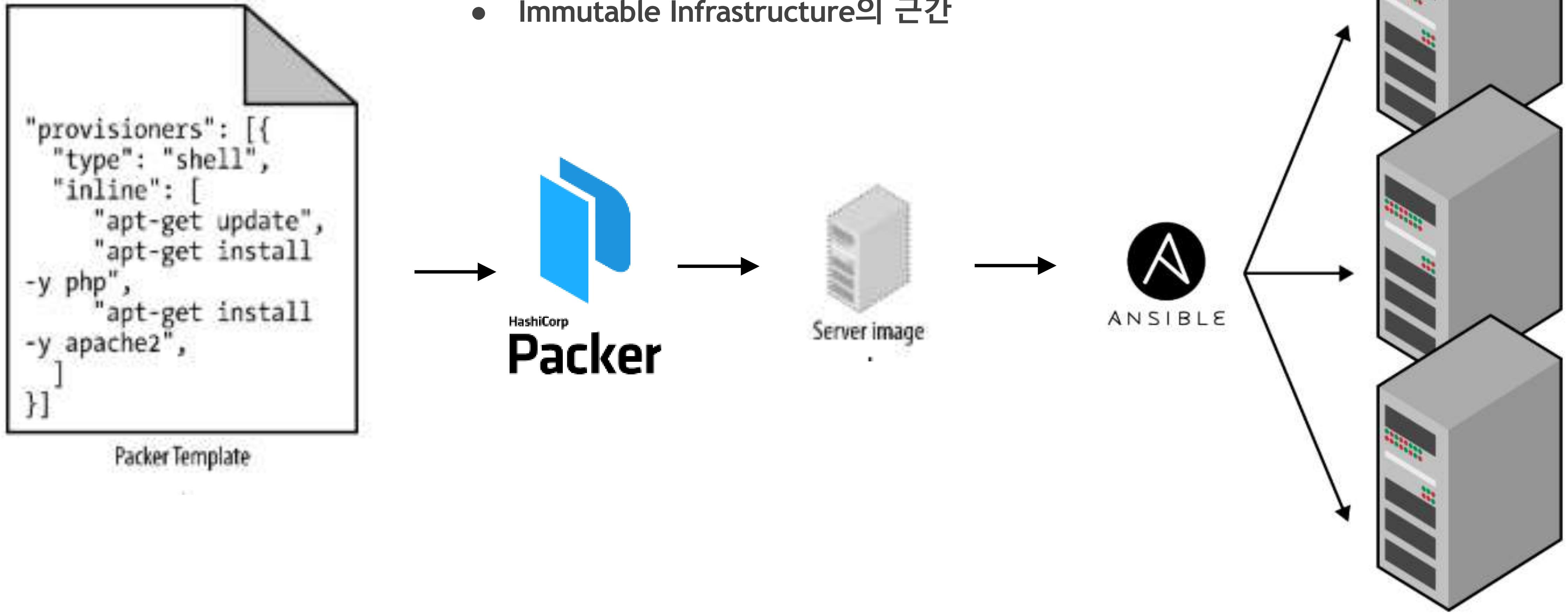


```
- name: Update cache
  apt:
    update_cache: yes

- name: Install PHP
  apt:
    name: php

- name: Install Apache
  apt:
    name: apache2
```

IaC Tools – Server Template Tools

Docker, Packer, Vagrant

- 서버 템플릿(소프트웨어 설치)을 이미지화
- Immutable Infrastructure의 근간

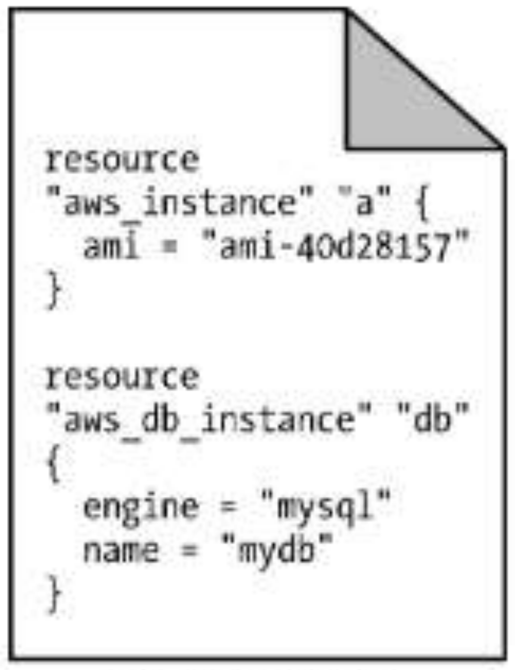


```
"provisioners": [{
  "type": "shell",
  "inline": [
     "apt-get update",
     "apt-get install
-y php",
     "apt-get install
-y apache2",
   ]
}]
```

IaC Tools – Provisioning Tools

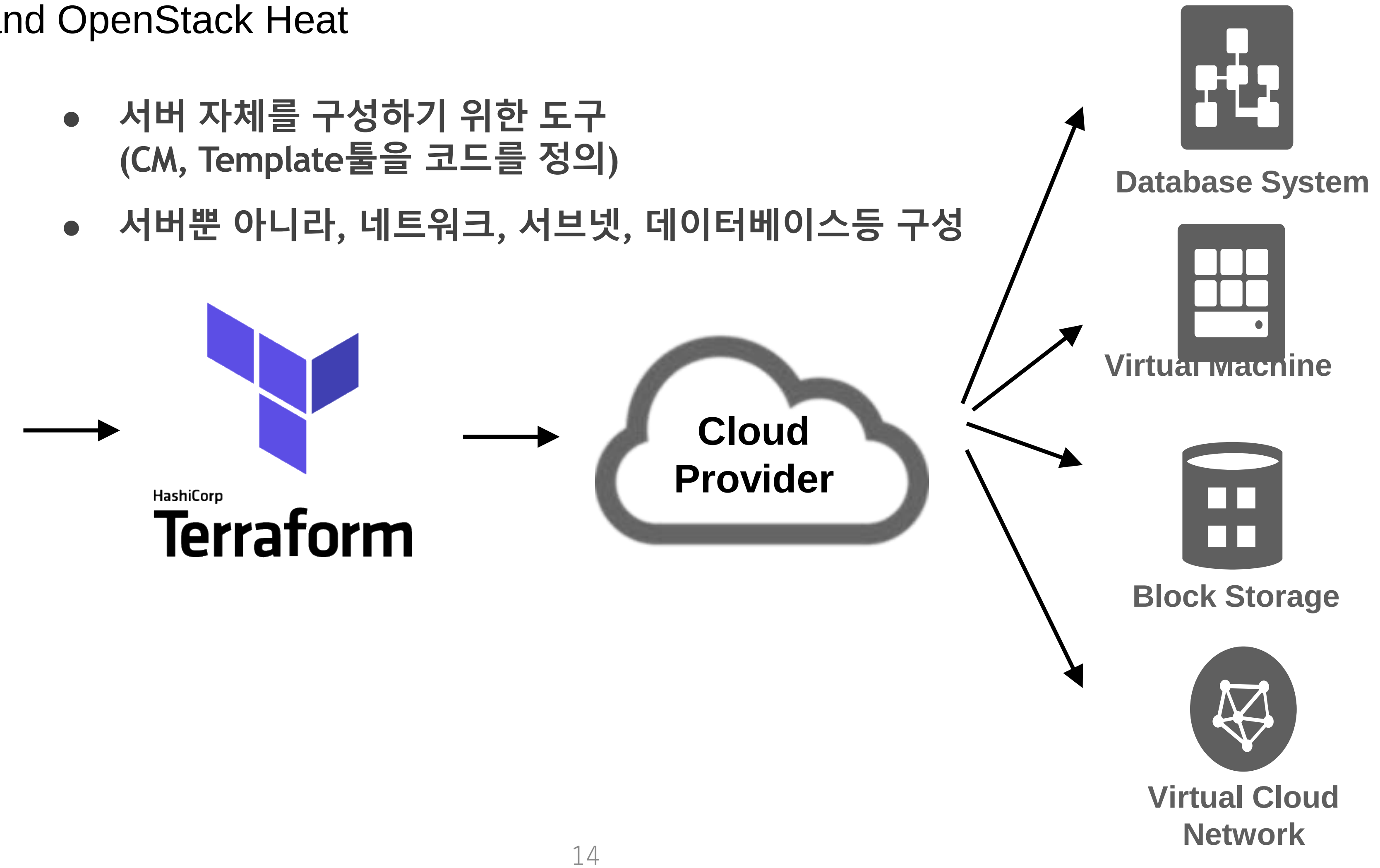
Terraform, Azure Resource Manager Templates,
AWS CloudFormation and OpenStack Heat

- 서버 자체를 구성하기 위한 도구
  (CM, Template툴을 코드를 정의)
- 서버뿐 아니라, 네트워크, 서브넷, 데이터베이스등 구성



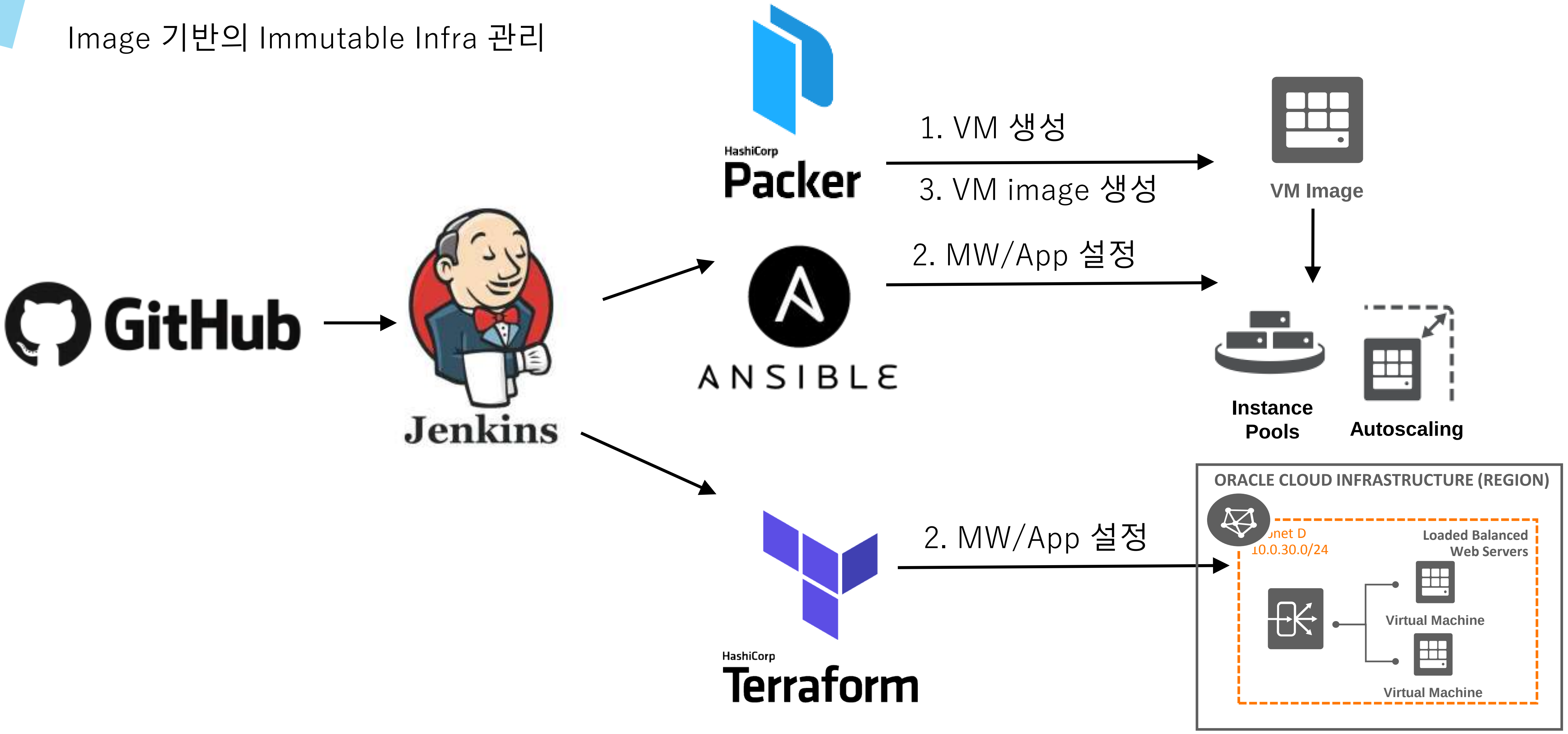
```
resource
"aws_instance" "a" {
  ami = "ami-40d28157"
}

resource
"aws_db_instance" "db"
{
  engine = "mysql"
  name = "mydb"
}
```

Terraform configuration



Immutable Infra Work Flow

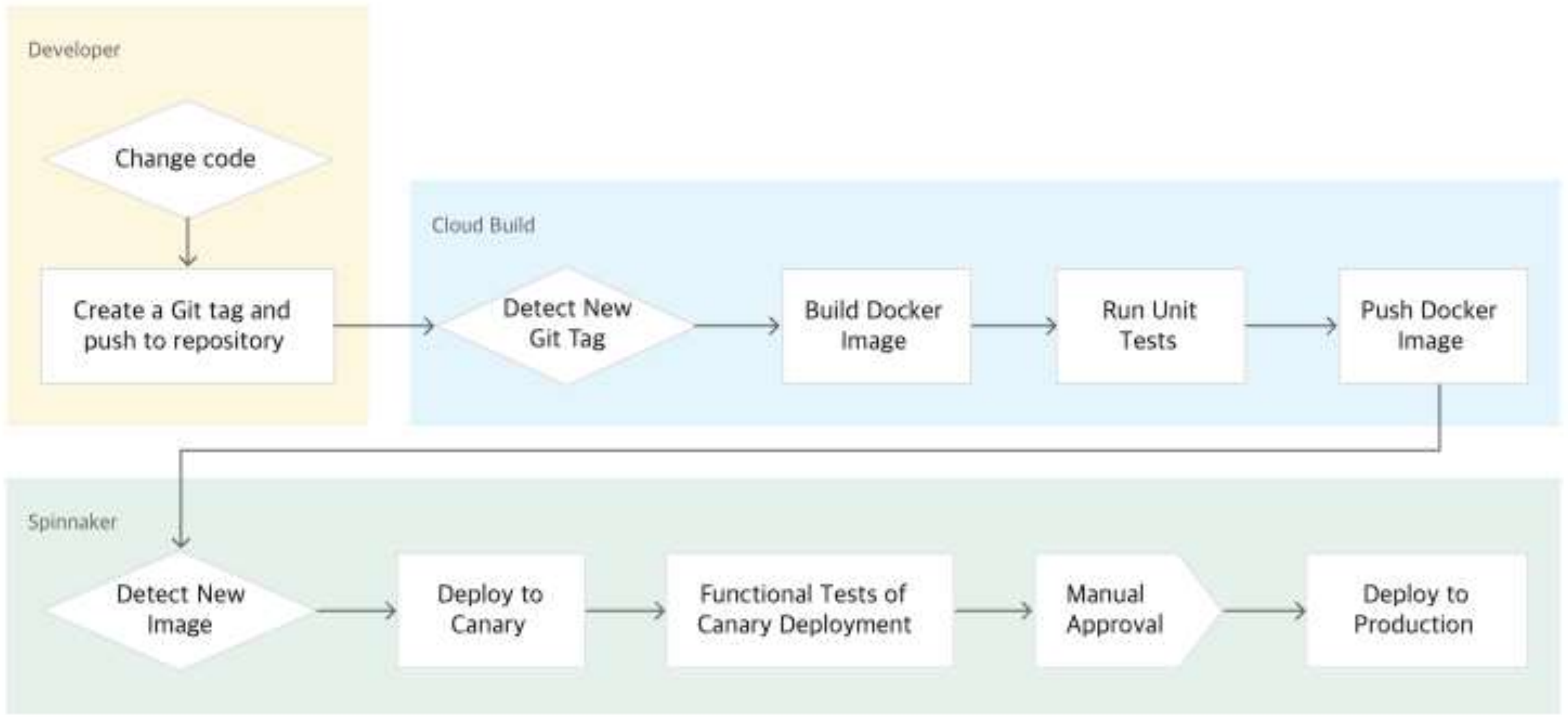
Image 기반의 Immutable Infra 관리



Demo Flow

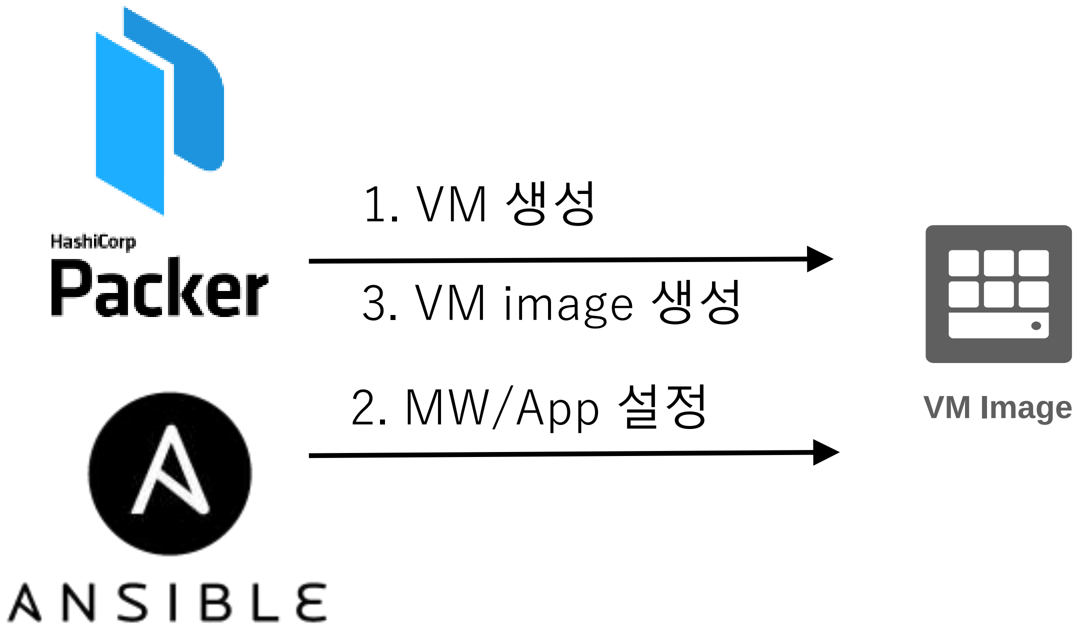
CD 툴을 이용한 전체 흐름

Spinnaker



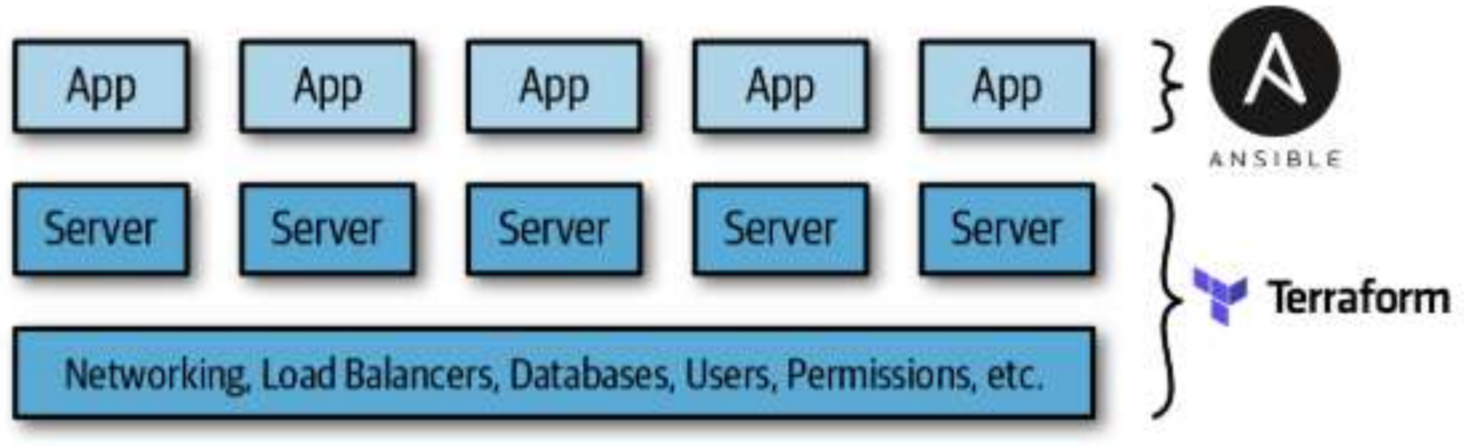
LIVE DEMO

LIVE DEMO

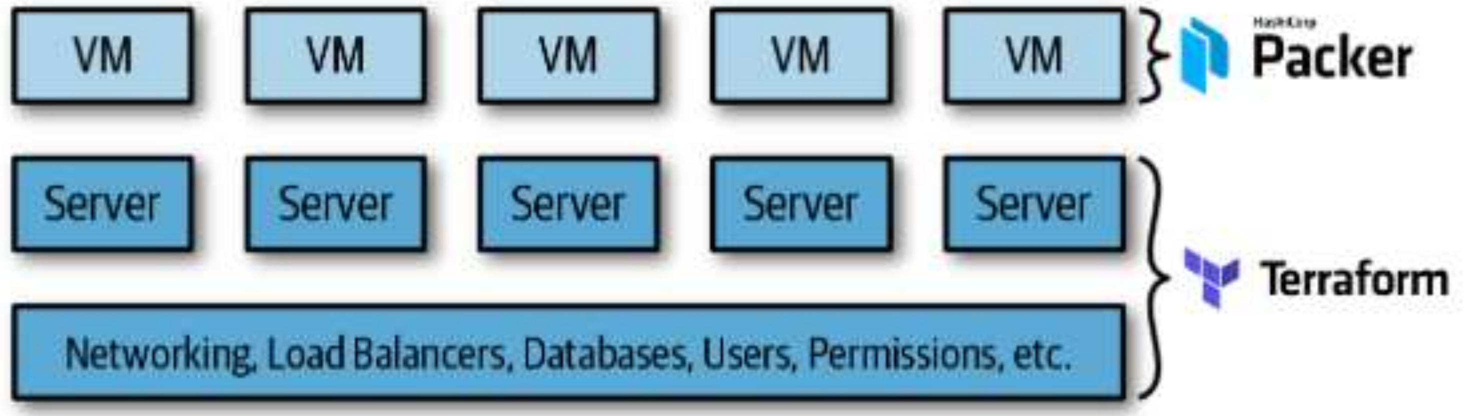

HashiCorp Packer
1. VM 생성
3. VM image 생성
2. MW/App 설정
ANSIBLE
VM Image

IaC Tools

Provisioning + configuration management


App
App
App
App
App
ANSIBLE
Server
Server
Server
Server
Server
Terraform
Networking, Load Balancers, Databases, Users, Permissions, etc.

Provisioning + server templating



IaC Tools – Comparison

Tool	Tool Type	Infrastructure	Architecture	Approach	Manifest Written Language
puppet	Configuration Management	Mutable	Pull	Declarative	Domain Specific Language (DSL) & Embedded Ruby (ERB)
CHEF	Configuration Management	Mutable	Pull	Declarative & Imperative	Ruby
ANSIBLE	Configuration Management	Mutable	Push	Declarative & Imperative	YAML
SALTSTACK	Configuration Management	Mutable	Push & Pull	Declarative & Imperative	YAML
Terraform	Provisioning	Immutable	Push	Declarative	HashiCorp Configuration Language (HCL)

Concepts – Imperative vs Declarative

Imperative(Procedural)

- imperative

  1. 필수의 2. 반드시 …해야 하는 3. 긴급한 4. 위엄 있는 5. 책무

  미국 [impérətiv] 영국 [impérətiv]

- 각각의 단계에 대한 **"특정 명령"을 일정한 순서에 따라 실행**해서 원하는 상태(Desired State) 에 이르게 한다.
- How 를 정의

1

집에서 나가라

2

차를 타라

3

영동대로에서
5키로 직진

4

5블럭 지나서 좌회전 후
500미터 전방에서 우회전

5

사거리 지나서 왼쪽에 아셈 타워

Concepts – Imperative vs Declarative

Declarative

- declarative

  1.단정적인 2.선언하는 3.서술적인

  미국 [diklǽrətiv] 영국 [diklǽrətiv]

- 원하는 상태(Desired State)를 정의 하고 시스템에 실행하게 하게 한다.

- What을 정의

대한민국 서울특별시 강남구
삼성1동 영동대로 517
아셈타워

Concepts – Mutable vs Immutable

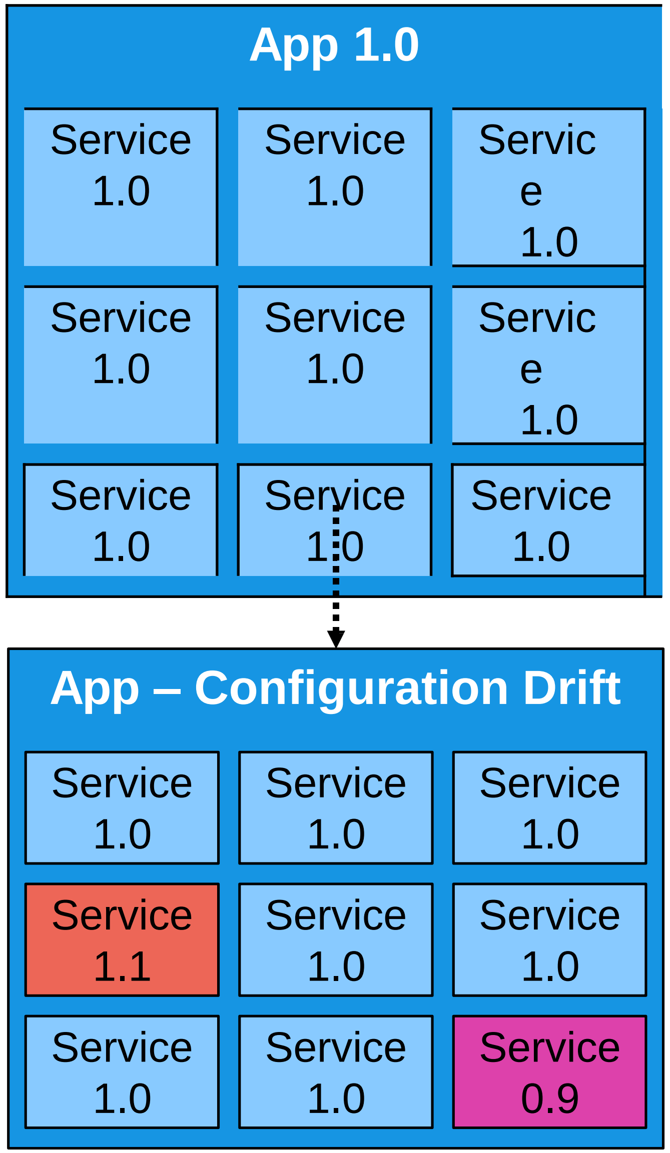
Mutable

- mutable

  1. 변덕스러운 2. 변하기 쉬운 3. 가변성의

  미국 [mjúːtəbl] 영국 [mjúːtəbl]

- CM Tools들은 기본적으로 Mutable (Ansible, Chef, Puppet등)
- 지속적으로 변경을 적용함에 따라 시간이 지나면 점점 설정의 불일치가 발생할 가능성 농후



Concepts – Mutable vs Immutable

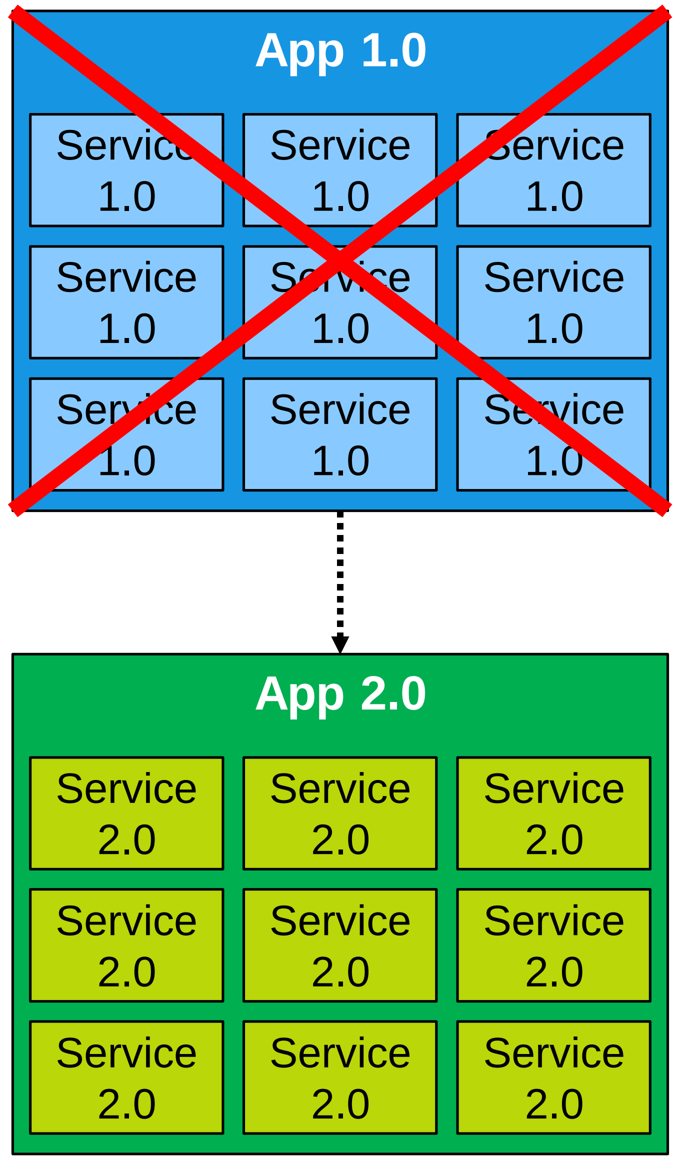
Immutable

- immutable

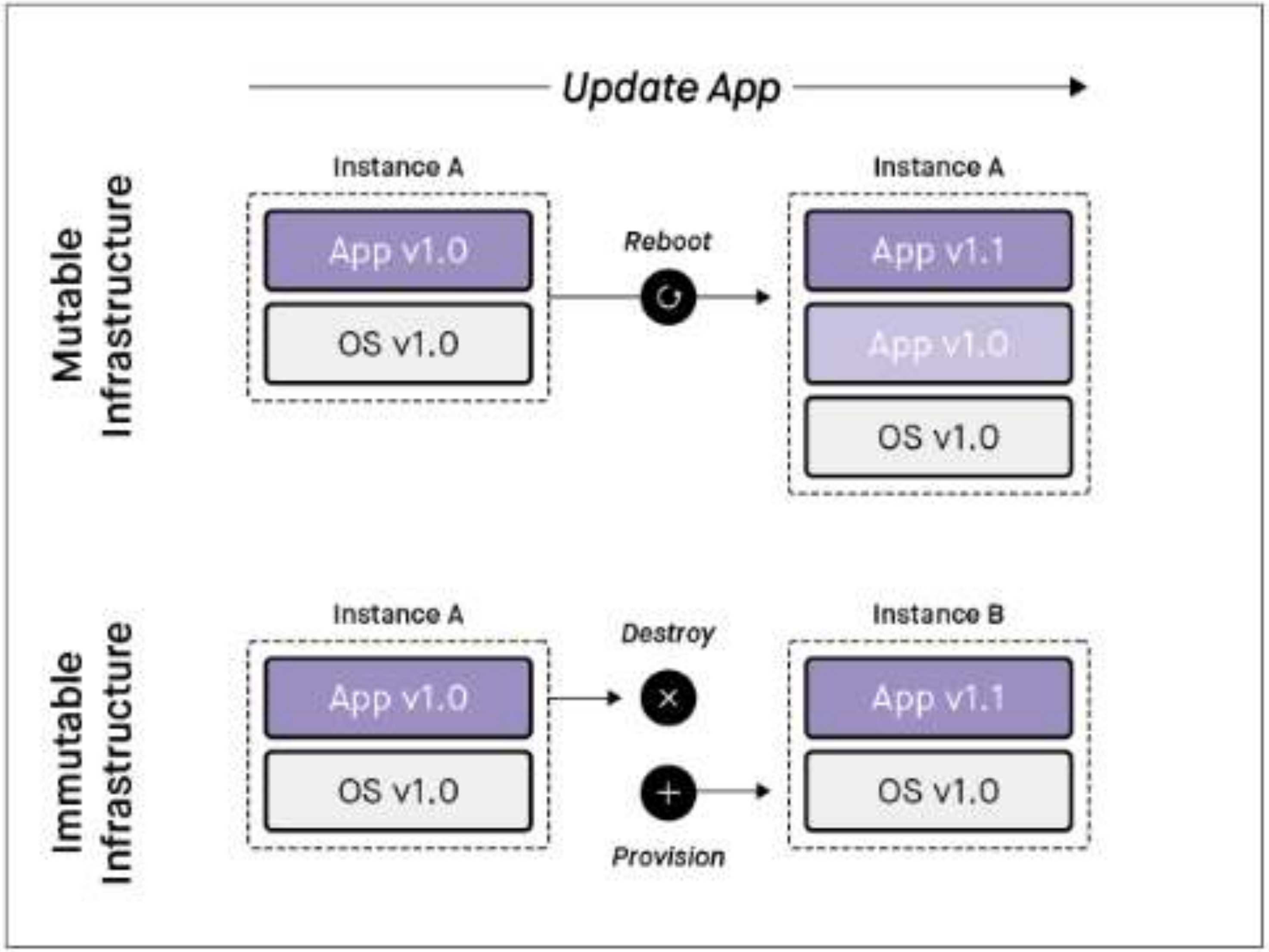
  1. 불변의 2. 바꿀 수 없는 3. 변경할 수 없는

  미국 [imjúːtəbl] 영국 [imjúːtəbl]

- Terraform과 같은 Provisioning Tool을 활용해서 Docker 또는 Packer가 만들어 놓은 이미지를 기반으로 완전히 새로운 인프라 생성
- 서버 간의 설정 불일치의 가능성을 제거
- 테스트를 통과한 Deployment를 운영환경으로 변경
- Blue/Green Deployment

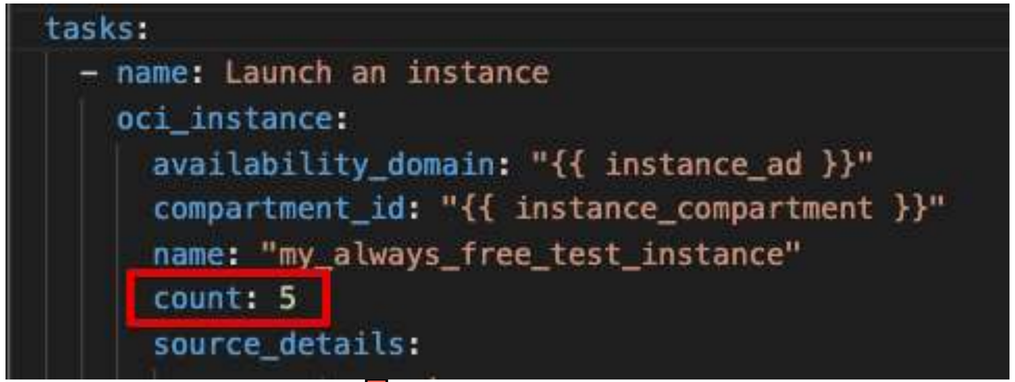


Concepts – Mutable vs Immutable



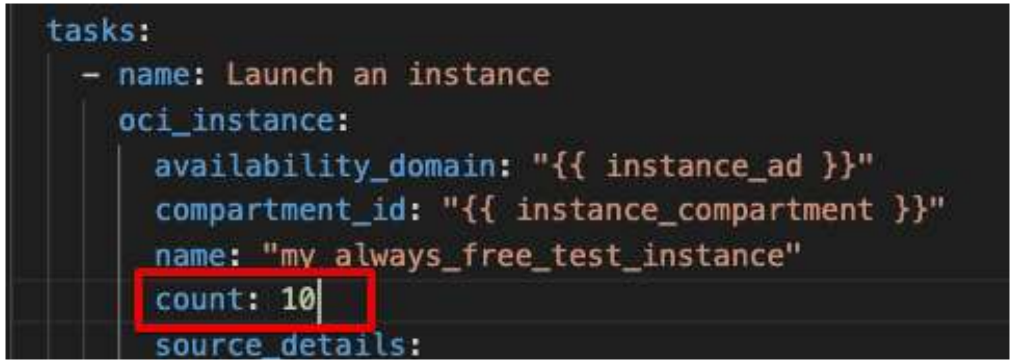
Concepts – Mutable vs Immutable

Example: Ansible Playbook



```
tasks:
  - name: Launch an instance
    oci_instance:
      availability_domain: "{{ instance_ad }}"
      compartment_id: "{{ instance_compartment }}"
      name: "my_always_free_test_instance"
      count: 5
      source_details:
```

+

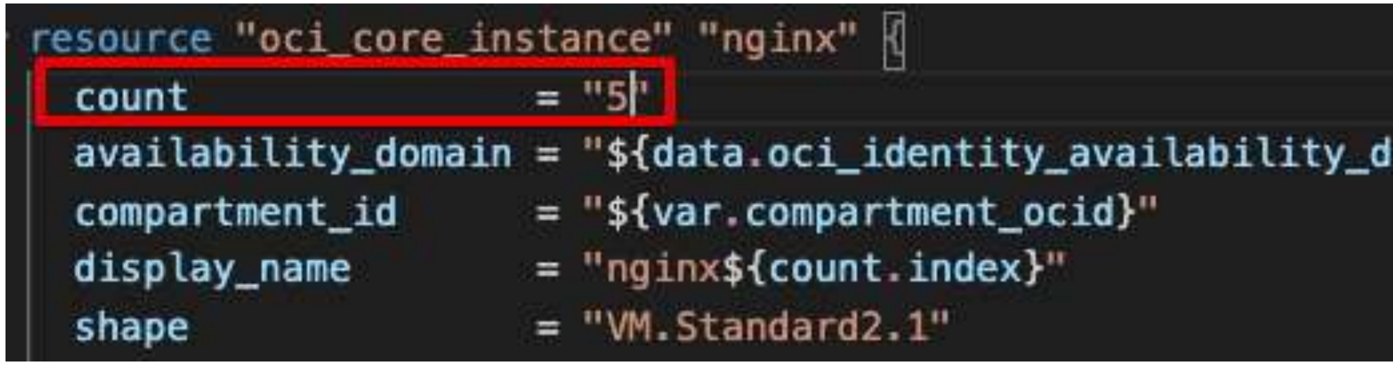


```
tasks:
  - name: Launch an instance
    oci_instance:
      availability_domain: "{{ instance_ad }}"
      compartment_id: "{{ instance_compartment }}"
      name: "my_always_free_test_instance"
      count: 10
      source_details:
```

=

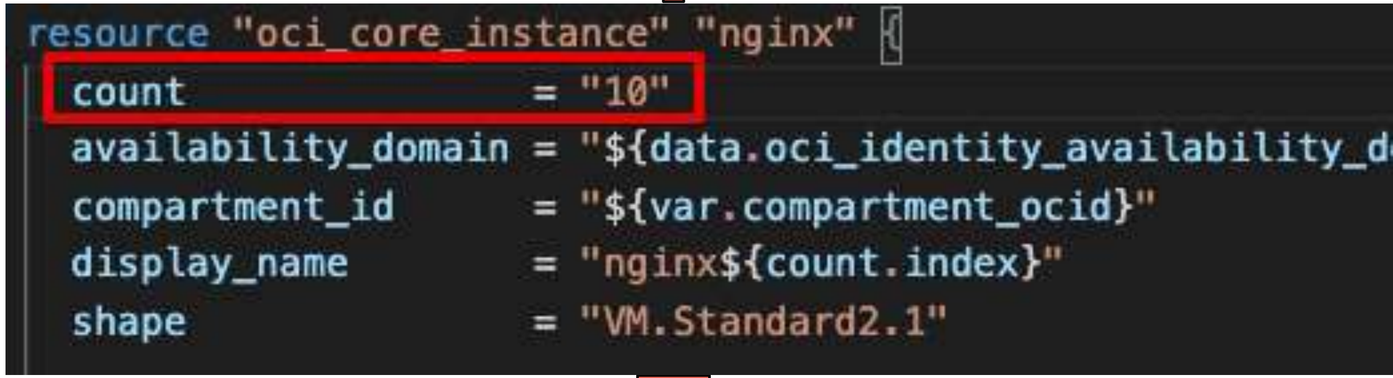
15 Servers

Example: Terraform Plan



```
resource "oci_core_instance" "nginx" {
  count               = "5"
  availability_domain = "${data.oci_identity_availability_d
  compartment_id      = "${var.compartment_ocid}"
  display_name        = "nginx${count.index}"
  shape               = "VM.Standard2.1"
```

+



```
resource "oci_core_instance" "nginx" {
  count               = "10"
  availability_domain = "${data.oci_identity_availability_d
  compartment_id      = "${var.compartment_ocid}"
  display_name        = "nginx${count.index}"
  shape               = "VM.Standard2.1"
```

=

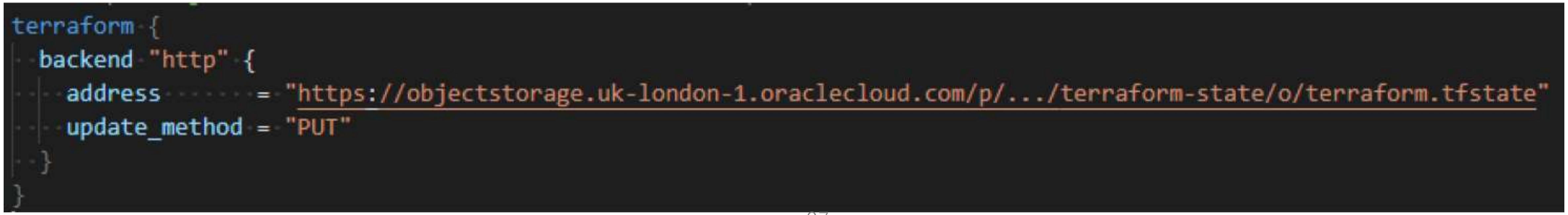
10 Servers

Terraform State

TF State

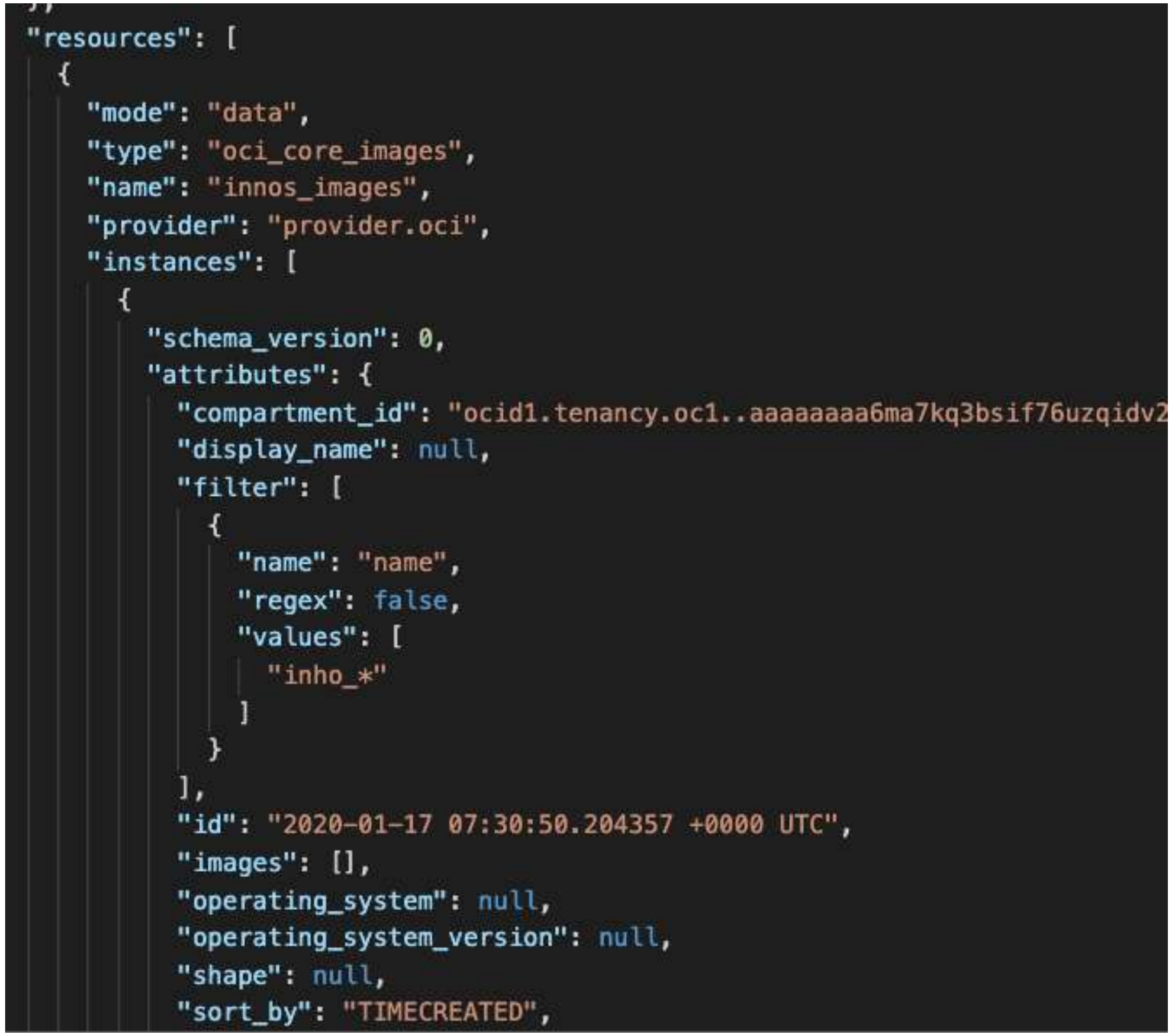
- 테라폼은 리소스의 상태 정보를 저장 terraform.tfstate에 저장
- 이 state 파일을 통해 리소스의 현재 상태와 tf 스크립트에 있는 설정 정보를 추적할 수 있다.
- 기본적으로 local 파일 (terraform.tfstate)에 저장되며, 원격지에 저장할 수 있는 기능 제공(backend)
- Backend를 사용하면 여러 팀에 상태정보를 공유할 수 있기 때문에 협업을 향상 시키지만 부수적인 문제점 발생
    - Lock, Encryption, Version 등 문제

HashiCorp Consul etcd Terraform ENTERPRISE ORACLE Cloud Infrastructure Object Storage AWS S3 Azure Blob GCP Storage

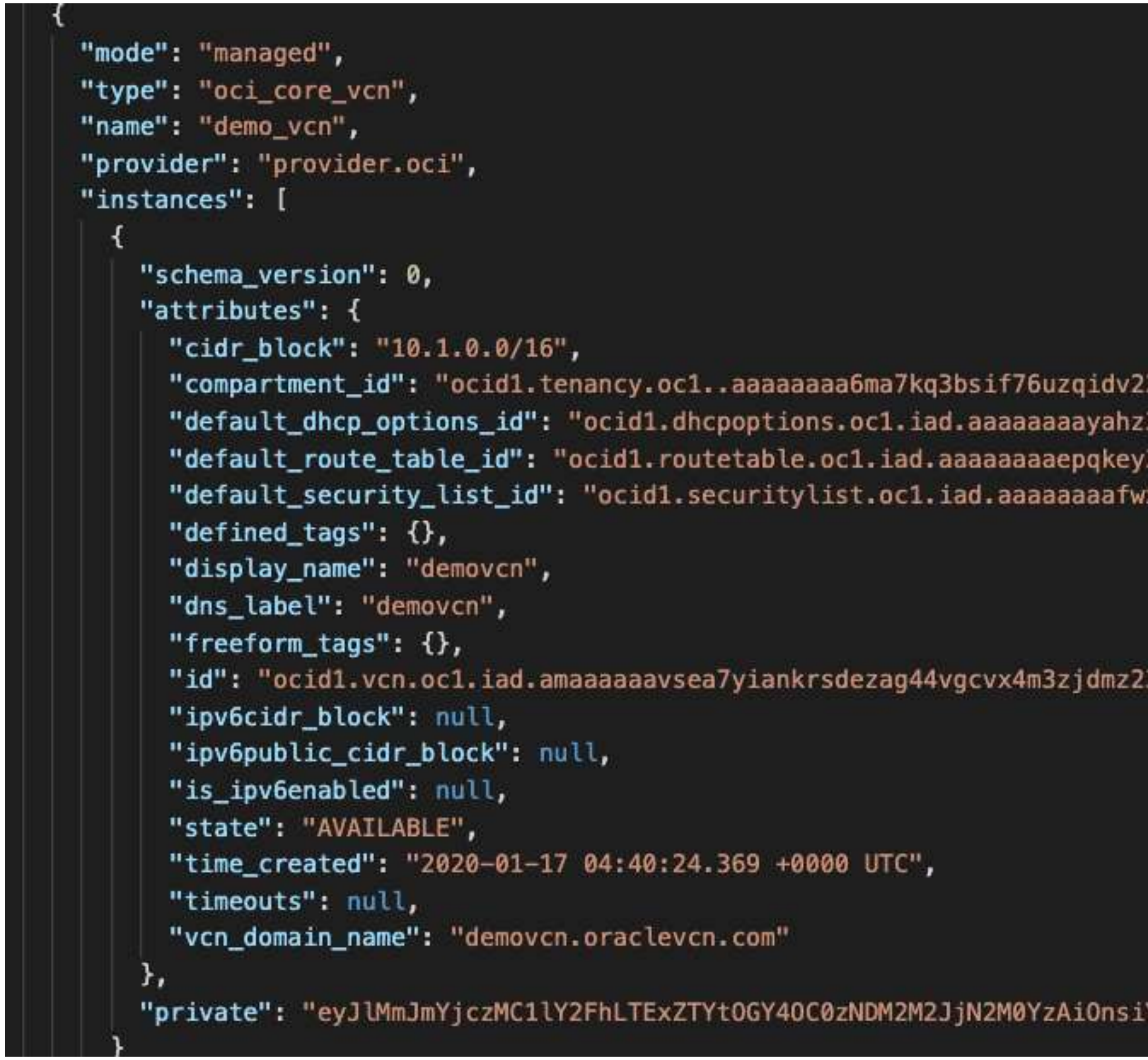


```
terraform {
  backend "http" {
    address       = "https://objectstorage.uk-london-1.oraclecloud.com/p/.../terraform-state/o/terraform.tfstate"
    update_method = "PUT"
  }
}
```

Terraform State

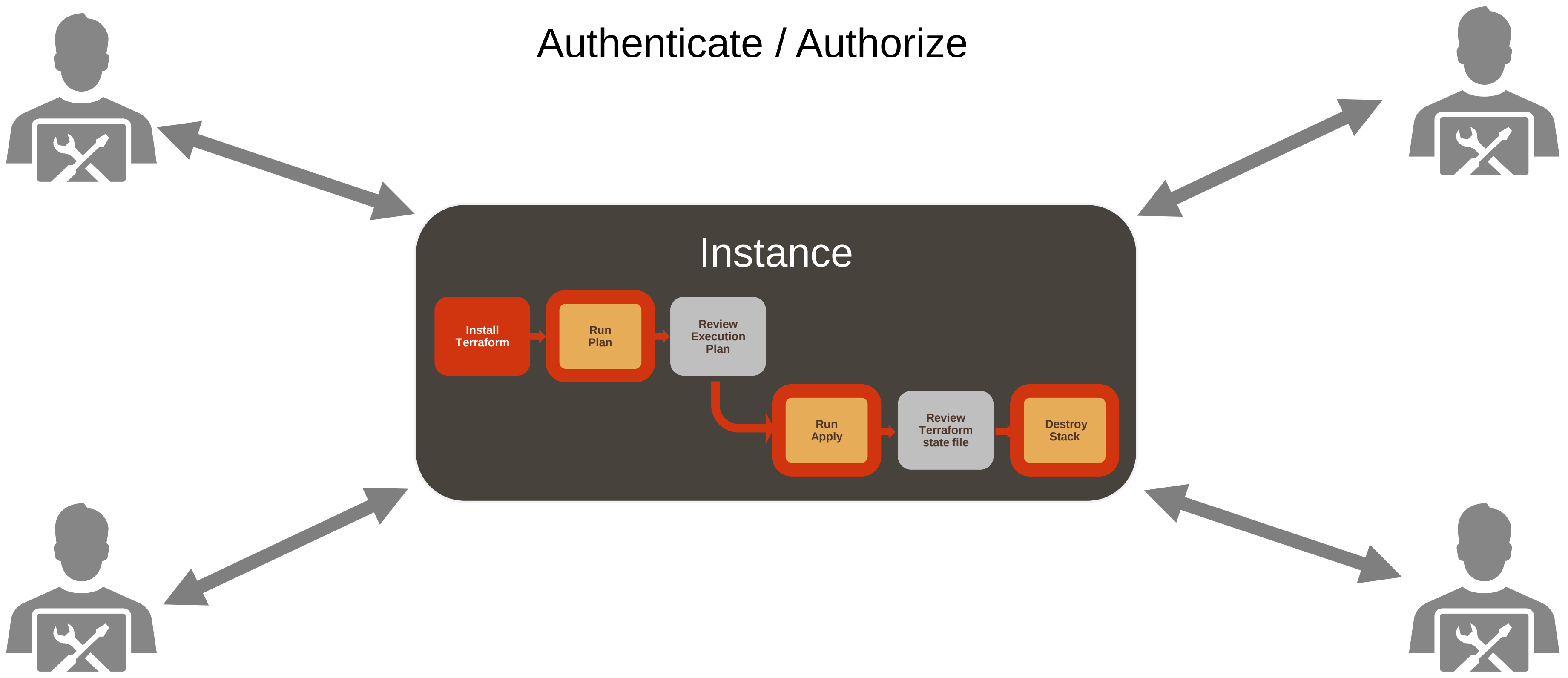


```
},
"resources": [
  {
    "mode": "data",
    "type": "oci_core_images",
    "name": "innos_images",
    "provider": "provider.oci",
    "instances": [
      {
        "schema_version": 0,
        "attributes": {
          "compartment_id": "ocid1.tenancy.oc1..aaaaaaaa6ma7kq3bsif76uzqidv2
          "display_name": null,
          "filter": [
            {
              "name": "name",
              "regex": false,
              "values": [
                "inho_*"
              ]
            }
          ],
          "id": "2020-01-17 07:30:50.204357 +0000 UTC",
          "images": [],
          "operating_system": null,
          "operating_system_version": null,
          "shape": null,
          "sort_by": "TIMECREATED",
```



```
{
  "mode": "managed",
  "type": "oci_core_vcn",
  "name": "demo_vcn",
  "provider": "provider.oci",
  "instances": [
    {
      "schema_version": 0,
      "attributes": {
        "cidr_block": "10.1.0.0/16",
        "compartment_id": "ocid1.tenancy.oc1..aaaaaaaa6ma7kq3bsif76uzqidv2
        "default_dhcp_options_id": "ocid1.dhcpoptions.oc1.iad.aaaaaaaayahz
        "default_route_table_id": "ocid1.routetable.oc1.iad.aaaaaaaaepqkey
        "default_security_list_id": "ocid1.securitylist.oc1.iad.aaaaaaaafw
        "defined_tags": {},
        "display_name": "demovcn",
        "dns_label": "demovcn",
        "freeform_tags": {},
        "id": "ocid1.vcn.oc1.iad.amaaaaaavsea7yiankrsdezag44vgcvx4m3zjdmz2
        "ipv6cidr_block": null,
        "ipv6public_cidr_block": null,
        "is_ipv6enabled": null,
        "state": "AVAILABLE",
        "time_created": "2020-01-17 04:40:24.369 +0000 UTC",
        "timeouts": null,
        "vcn_domain_name": "demovcn.oraclevcn.com"
      },
      "private": "eyJlMmJmYjczMC1lY2FhLTExZTYtOGY4OC0zNDM2M2JjN2M0YzAiOnsi
    }
```

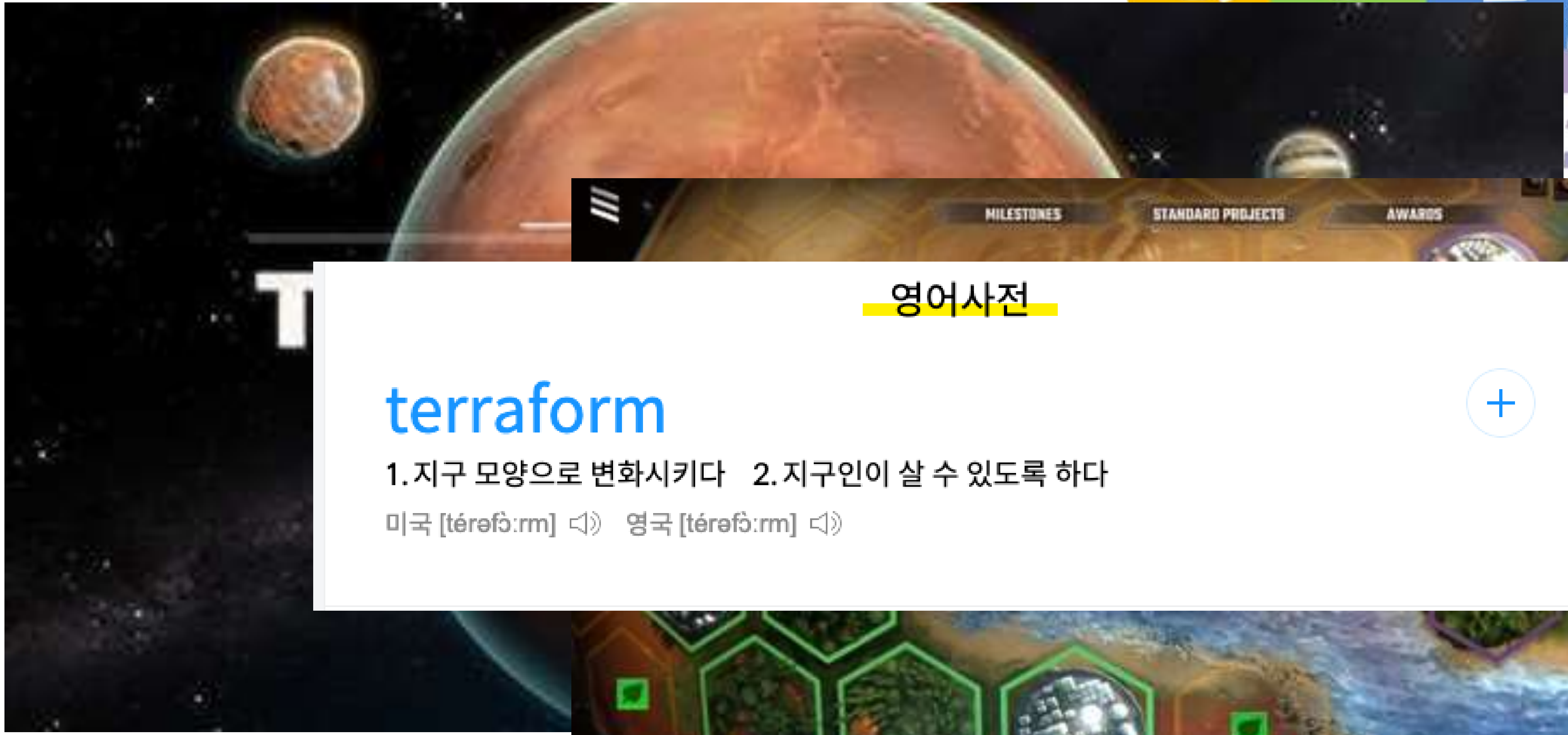
Resource Manager

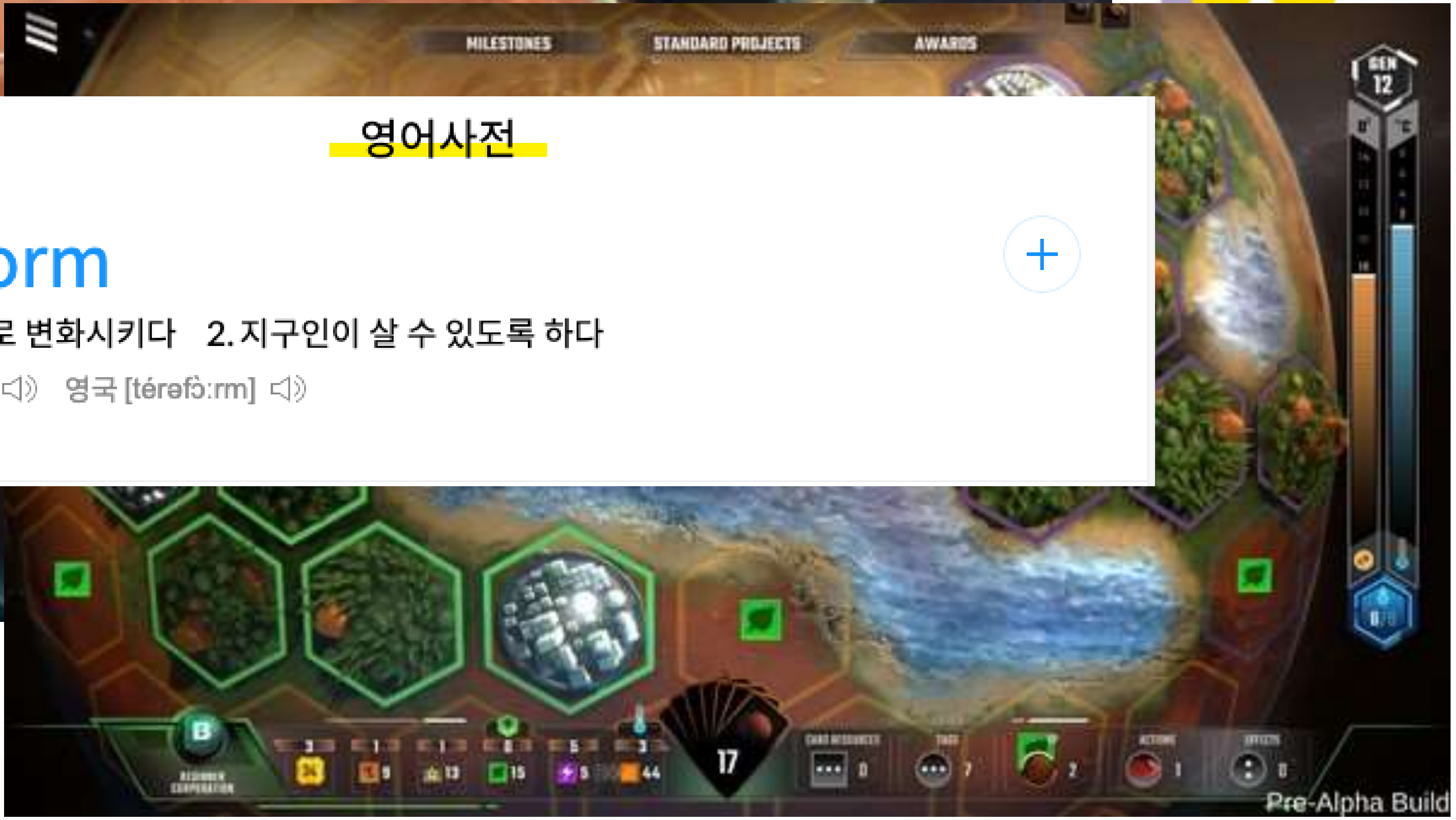


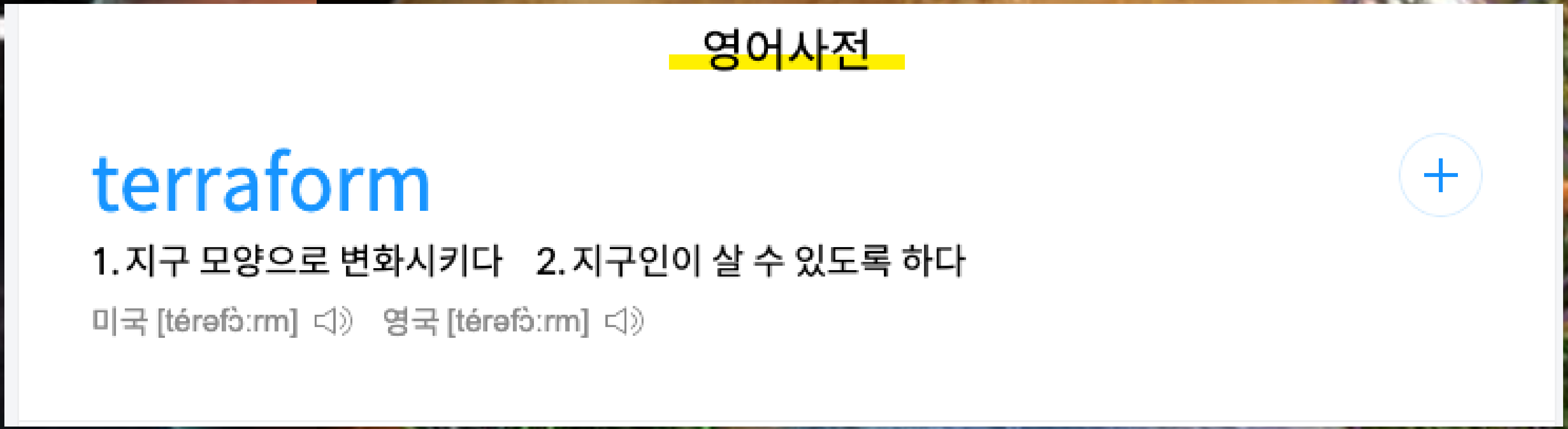
PART 2

Terraform

Terraform







영어사전

terraform

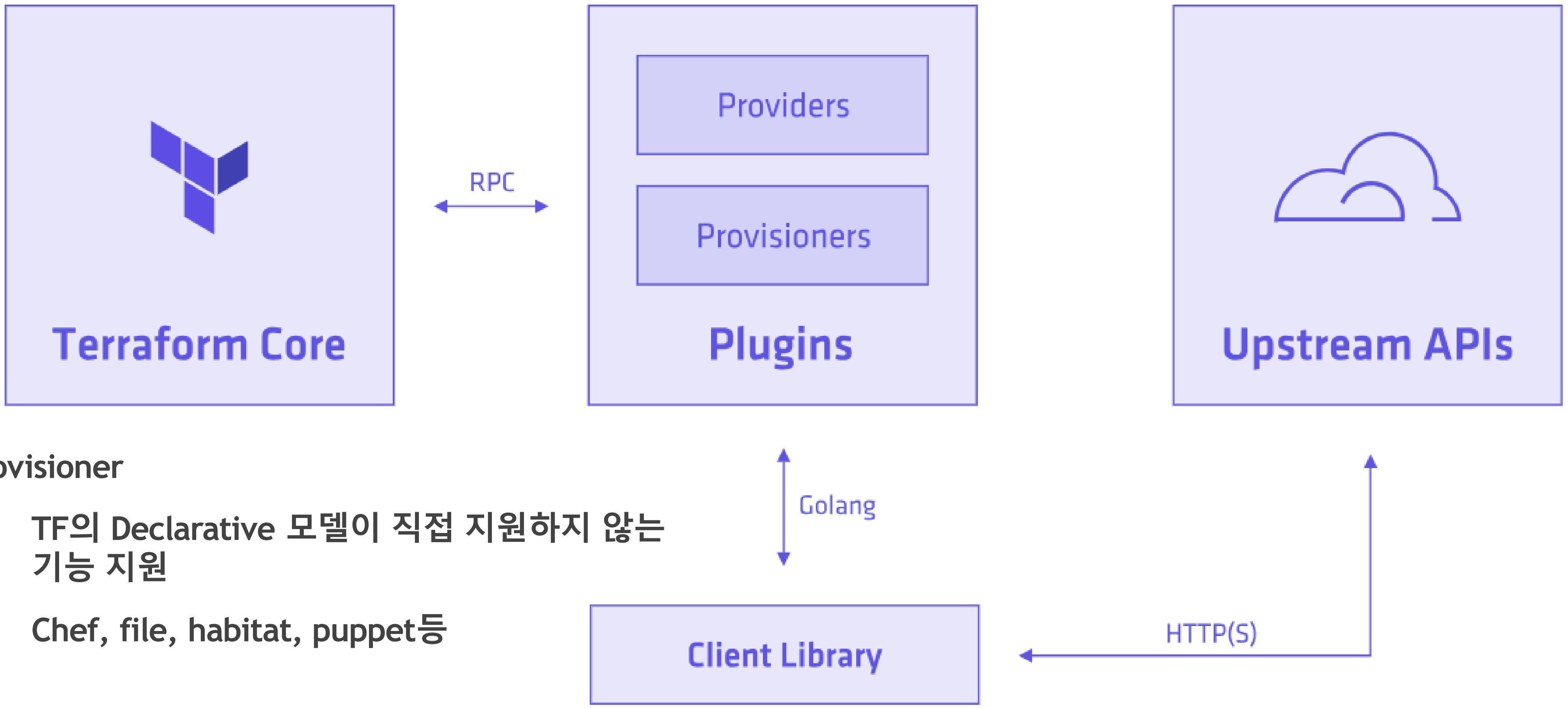
1. 지구 모양으로 변화시키다 2. 지구인이 살 수 있도록 하다

미국 [térəfɔ̀:rm] 영국 [térəfɔ̀:rm]

Terraform

- A provisioning declarative tool that based on Infrastructure as a Code paradigm
- HCL (Hashicorp Configuration Language)
- Written in Golang.
- Helps to evolve you infrastructure, safely and predictably
- Applies Graph Theory to IaaC
- Terraform is a multipurpose composition tool:
  - Composes multiple tiers (SaaS/PaaS/IaaS)
  - A plugin-based architecture model
- Open source. Backed by Hashicorp company (Guide/Principles/Design)

Terraform Architecture



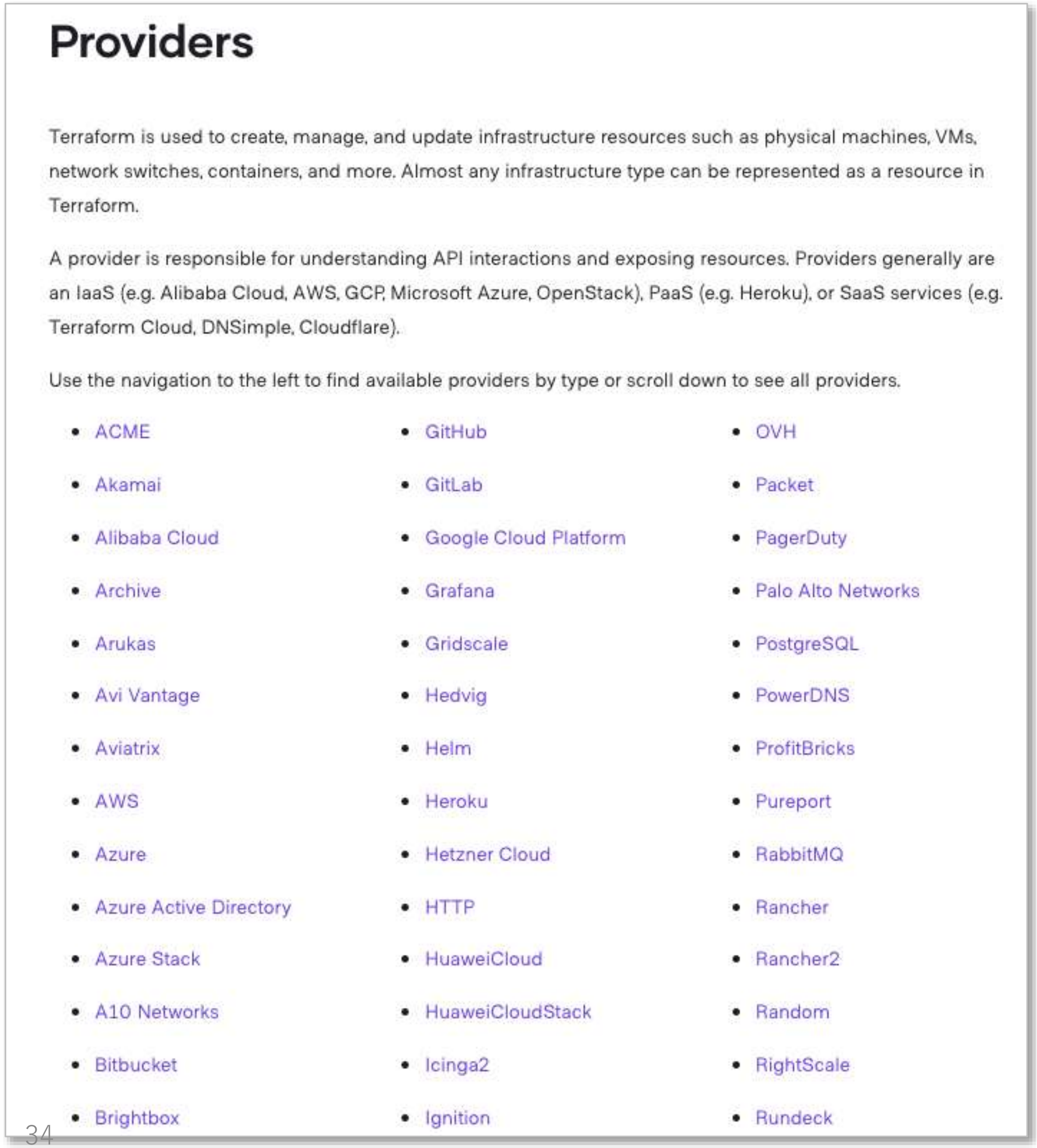
- Provisioner
  - TF의 Declarative 모델이 직접 지원하지 않는 기능 지원
  - Chef, file, habitat, puppet등

Terraform Core Concept #1

Providers

- 테라폼과 외부의 리소스간에 API를 통신을 담당
- 다양한 외부 프로바이더
  - IaaS (ex: OCI, AWS, GCP, Microsoft Azure)
  - PaaS (ex: Heroku),
  - SaaS services (ex: Terraform Enterprise, DNSimple, CloudFlare)

https://www.terraform.io/docs/providers/index.html



Providers

Terraform is used to create, manage, and update infrastructure resources such as physical machines, VMs, network switches, containers, and more. Almost any infrastructure type can be represented as a resource in Terraform.

A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. Alibaba Cloud, AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g. Terraform Cloud, DNSimple, Cloudflare).

Use the navigation to the left to find available providers by type or scroll down to see all providers.

- ACME
- Akamai
- Alibaba Cloud
- Archive
- Arukas
- Avi Vantage
- Aviatrix
- AWS
- Azure
- Azure Active Directory
- Azure Stack
- A10 Networks
- Bitbucket
- Brightbox
- GitHub
- GitLab
- Google Cloud Platform
- Grafana
- Gridscale
- Hedvig
- Helm
- Heroku
- Hetzner Cloud
- HTTP
- HuaweiCloud
- HuaweiCloudStack
- Icinga2
- Ignition
- OVH
- Packet
- PagerDuty
- Palo Alto Networks
- PostgreSQL
- PowerDNS
- ProfitBricks
- Pureport
- RabbitMQ
- Rancher
- Rancher2
- Random
- RightScale
- Rundeck

Terraform Core Concept #1

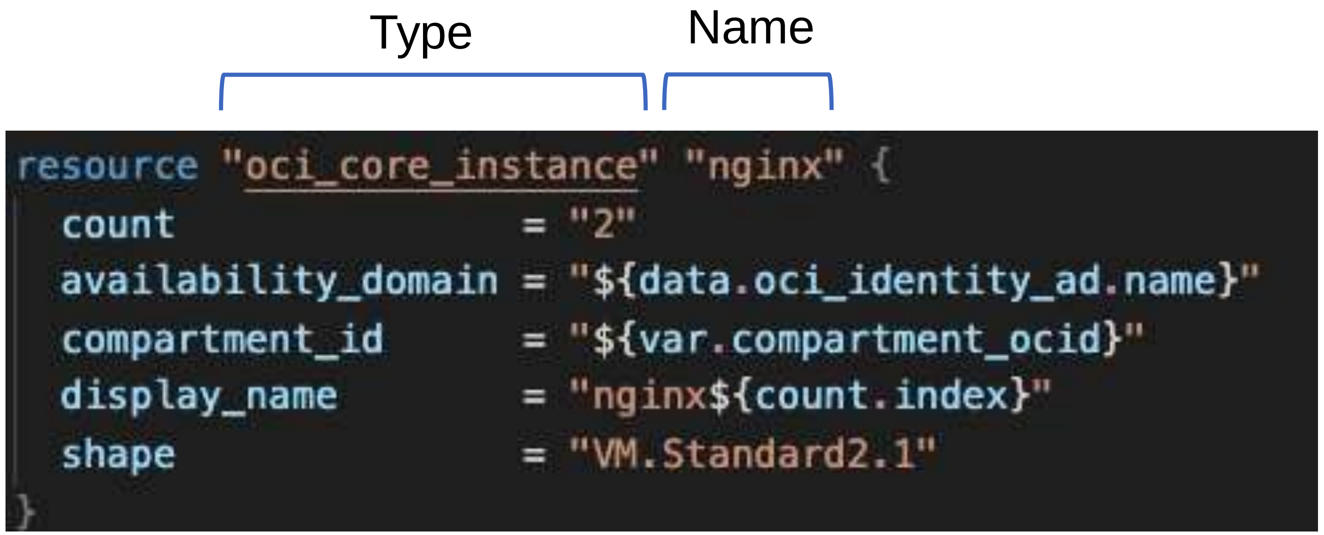
Providers

```
provider "oci" {
    tenancy_ocid = "${var.tenancy_ocid}"
    user_ocid = "${var.user_ocid}"
    fingerprint = "${var.fingerprint}"
    private_key_path = "${var.private_key_path}"
    region = "${var.region}"
}
```

Terraform Core Concept #2

Resource

- 리소스란 특정 프로바이더가 제공해주는 조작 가능한 대상의 최소 단위
- 예를 들면 OCI 프로바이더의 oci_core_instance는 VM을 만드는데 사용되는 리소스 타입
- 모든 클라우드의 자원이 "Resource"로 관리됨
- 테라폼은 이런 리소스의 생성, 관리, 갱신, 삭제(CRUD)을 담당

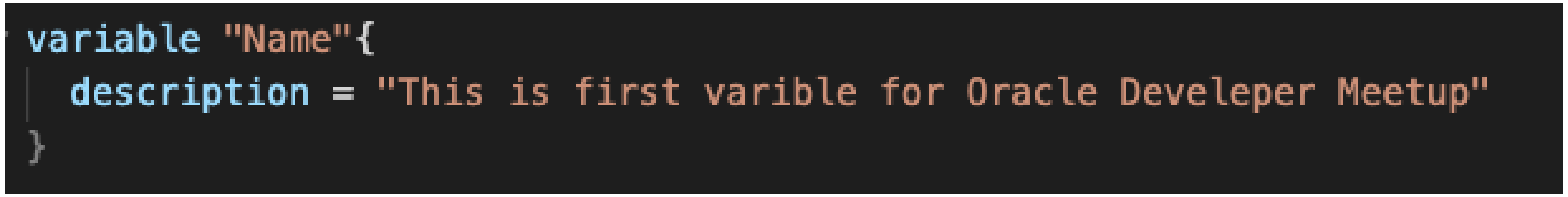


```
resource "oci_core_instance" "nginx" {
  count               = "2"
  availability_domain = "${data.oci_identity_ad.name}"
  compartment_id      = "${var.compartment_ocid}"
  display_name        = "nginx${count.index}"
  shape               = "VM.Standard2.1"
}
```

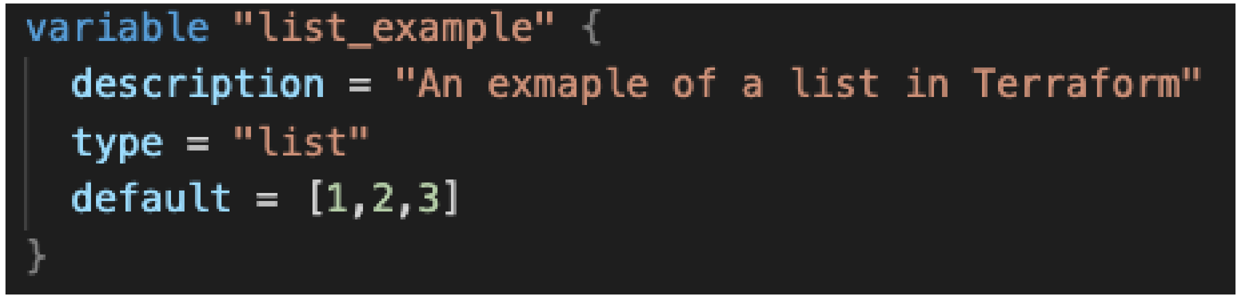
Terraform Core Concept #3

variable

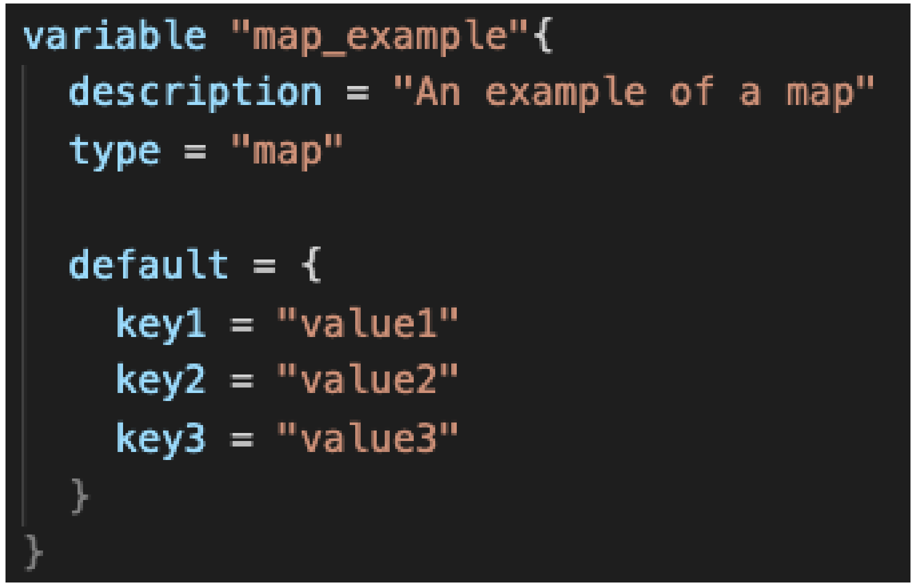
- 여러곳에 Hard Coding(ex:Port 번호)하는 것을 막기 위해 변수를 사용 (DRY, Don't Repeat Yourself)
- Input 변수에 값을 제공하는 방법은 1. -var 옵션, 2. --var-file 옵션, 3. TF_VAR_<변수명> 환경변수
- type은 string, list, map이며, 선언하지 않으면 string으로 간주



```
variable "Name"{
  description = "This is first varible for Oracle Develeper Meetup"
}
```



```
variable "list_example" {
  description = "An exmaple of a list in Terraform"
  type = "list"
  default = [1,2,3]
}
```

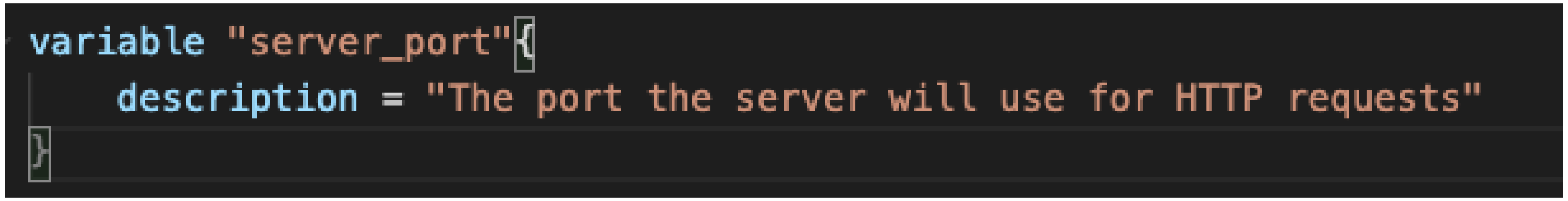


```
variable "map_example"{
  description = "An example of a map"
  type = "map"

  default = {
    key1 = "value1"
    key2 = "value2"
    key3 = "value3"
  }
}
```

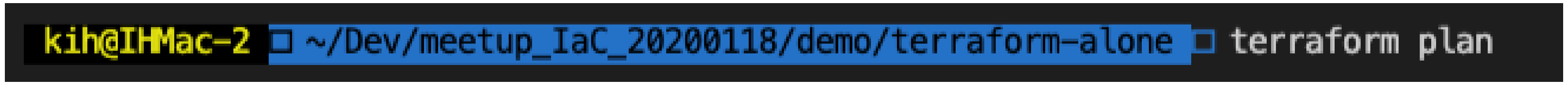
Terraform Core Concept #3

variable - 변수 입력 방법

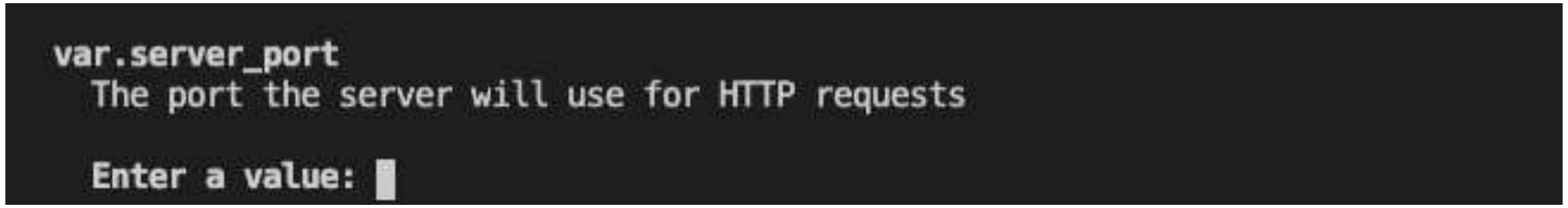
- Input 변수에 값을 제공하는 방법은 1. -var 옵션, 2. --var-file 옵션, 3. TF_VAR_<변수명> 환경변수



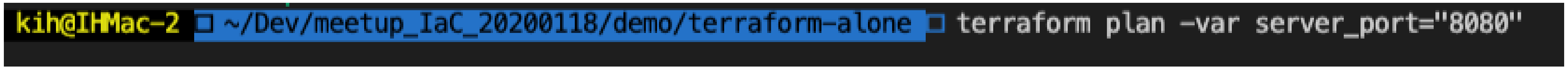
```
variable "server_port"{
    description = "The port the server will use for HTTP requests"
}
```



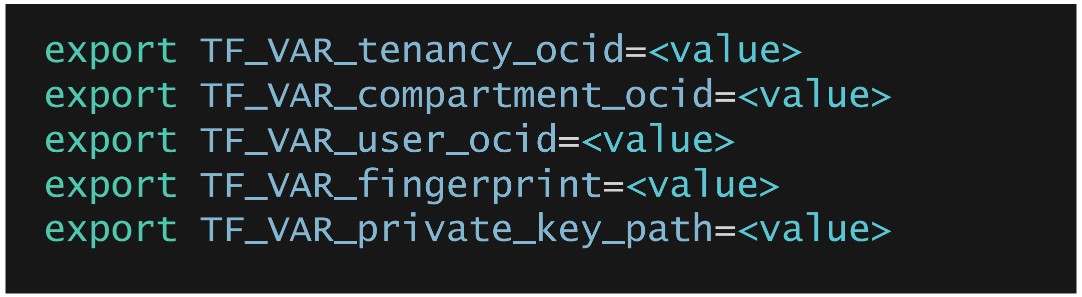
```
kih@IHMac-2 □ ~/Dev/meetup_IaC_20200118/demo/terraform-alone □ terraform plan
```



```
var.server_port
  The port the server will use for HTTP requests

  Enter a value:
```



```
kih@IHMac-2 □ ~/Dev/meetup_IaC_20200118/demo/terraform-alone □ terraform plan -var server_port="8080"
```

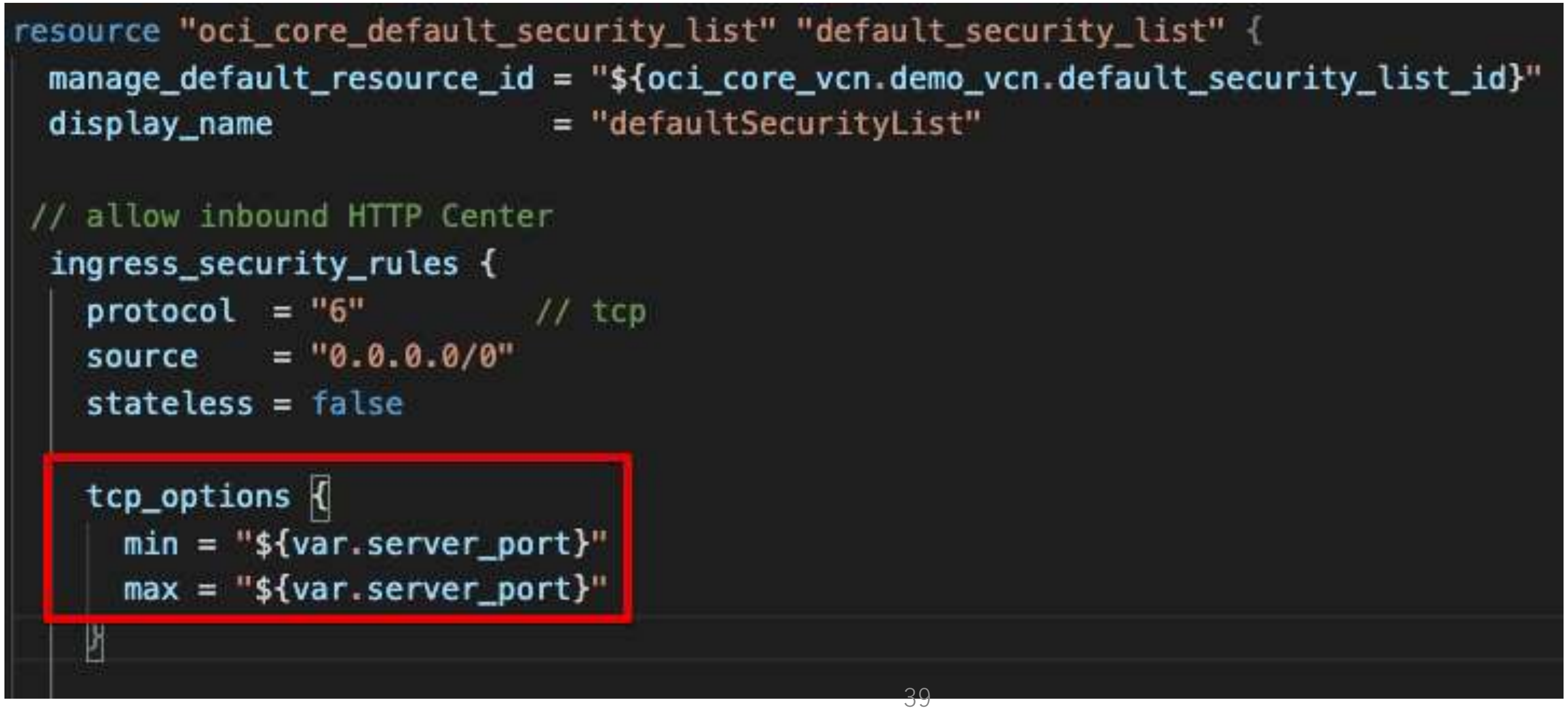


```
export TF_VAR_tenancy_ocid=<value>
export TF_VAR_compartment_ocid=<value>
export TF_VAR_user_ocid=<value>
export TF_VAR_fingerprint=<value>
export TF_VAR_private_key_path=<value>
```

Terraform Core Concept #3

variable - 사용 방법

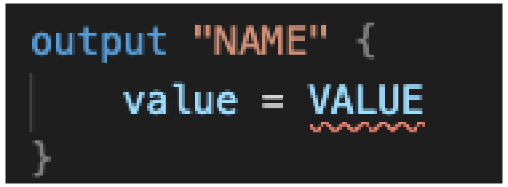
- ${var.<변수명>} (v0.11), var.<변수명> (v0.12)



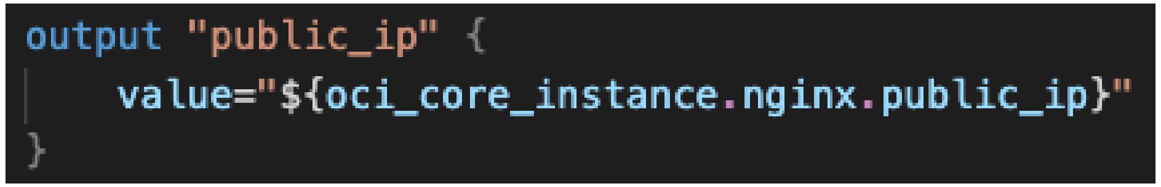
```
resource "oci_core_default_security_list" "default_security_list" {
  manage_default_resource_id = "${oci_core_vcn.demo_vcn.default_security_list_id}"
  display_name               = "defaultSecurityList"

 // allow inbound HTTP Center
  ingress_security_rules {
    protocol  = "6"         // tcp
    source    = "0.0.0.0/0"
    stateless = false

    tcp_options {
      min = "${var.server_port}"
      max = "${var.server_port}"
    }
```

Terraform Core Concept #3

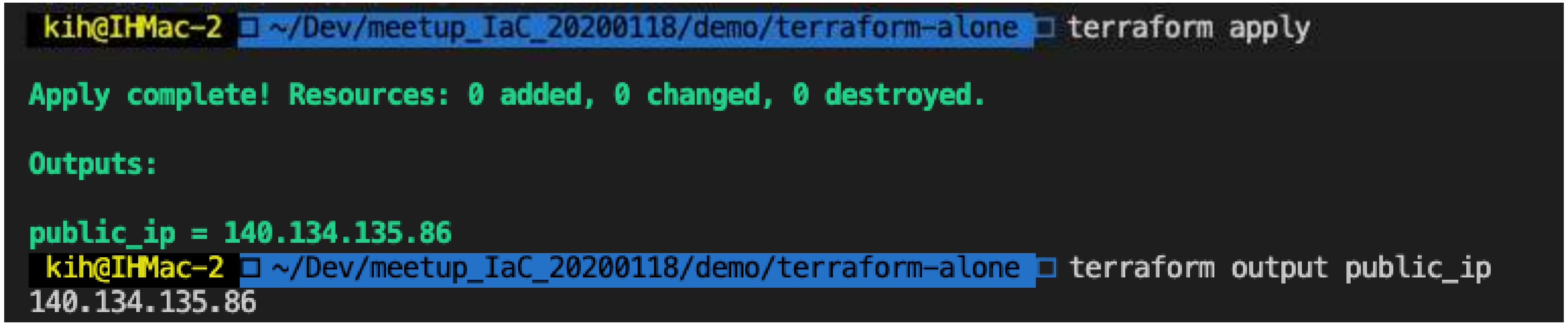
variable - output



```
output "NAME" {
    value = VALUE
}
```



```
output "public_ip" {
    value="${oci_core_instance.nginx.public_ip}"
}
```

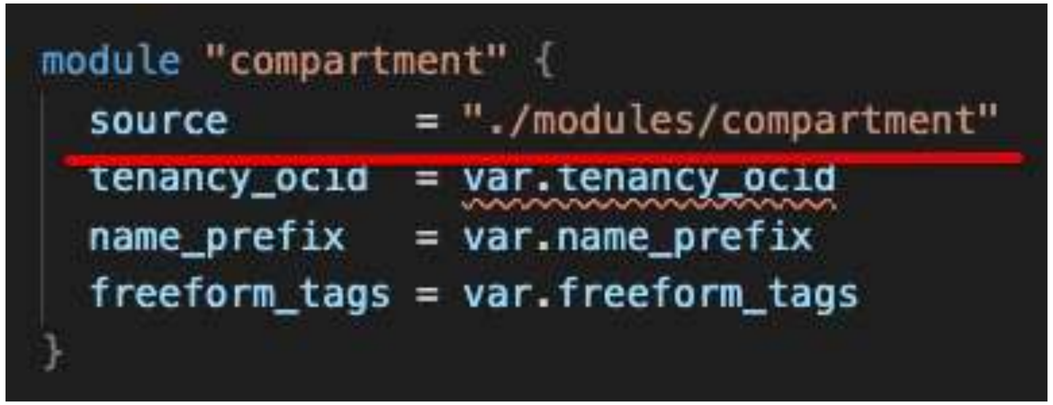


```
 kih@IHMac-2 □ ~/Dev/meetup_IaC_20200118/demo/terraform-alone □ terraform apply

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

Outputs:

public_ip = 140.134.135.86
 kih@IHMac-2 □ ~/Dev/meetup_IaC_20200118/demo/terraform-alone □ terraform output public_ip
140.134.135.86
```

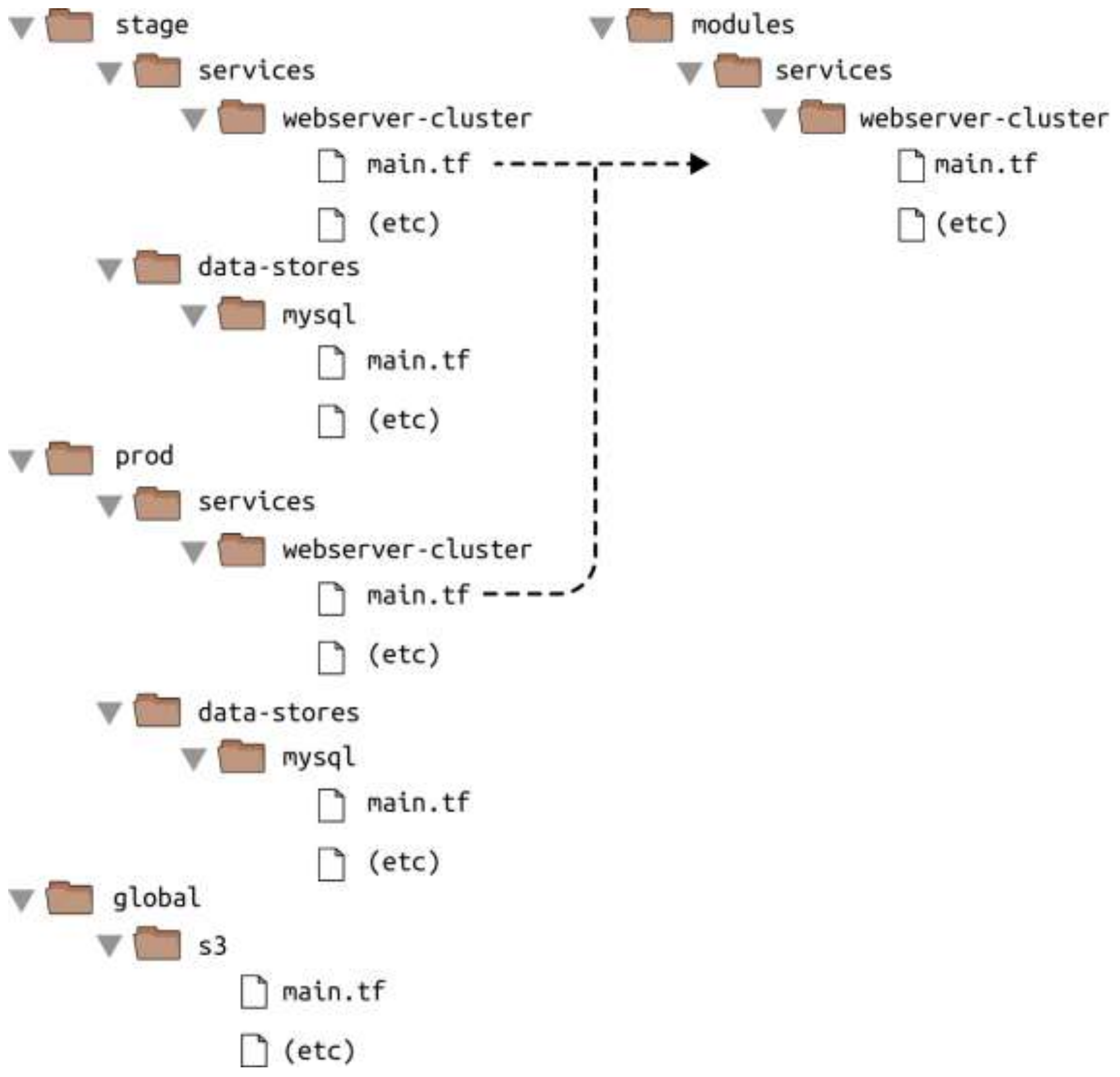
Terraform Core Concept #4

Module

- 모듈은 설정 파일을 그룹단위로 모듈화
- 여러 환경에서 재사용성을 재고
- 경로 설정은 상대경로로 설정
- Git등을 이용해 외부 저장소 활용



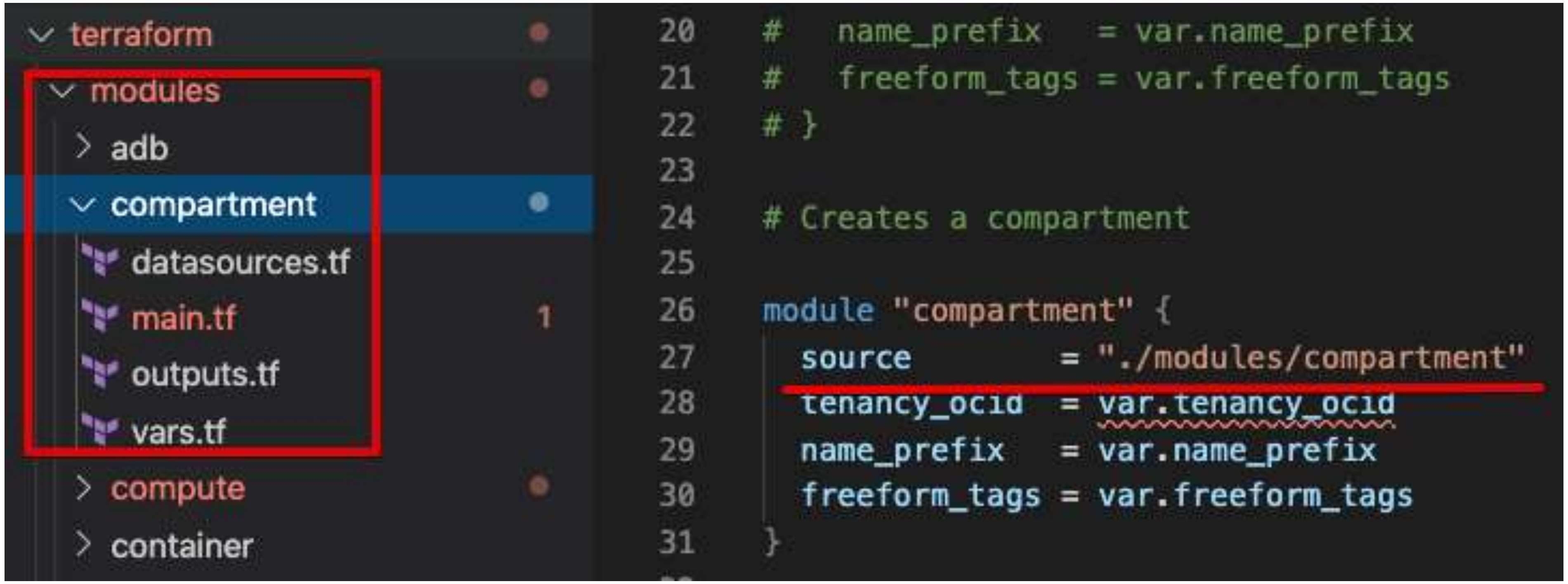
```
module "compartment" {
  source         = "./modules/compartment"
  tenancy_ocid   = var.tenancy_ocid
  name_prefix    = var.name_prefix
  freeform_tags  = var.freeform_tags
}
```



```
▼ stage                                  ▼ modules
  ▼ services                               ▼ services
    ▼ webserver-cluster                      ▼ webserver-cluster
        main.tf  ---------------------------->   main.tf
        (etc)                                    (etc)
  ▼ data-stores
    ▼ mysql
        main.tf
        (etc)
▼ prod
  ▼ services
    ▼ webserver-cluster
        main.tf
        (etc)
  ▼ data-stores
    ▼ mysql
        main.tf
        (etc)
▼ global
  ▼ s3
      main.tf
      (etc)
```

Terraform Core Concept

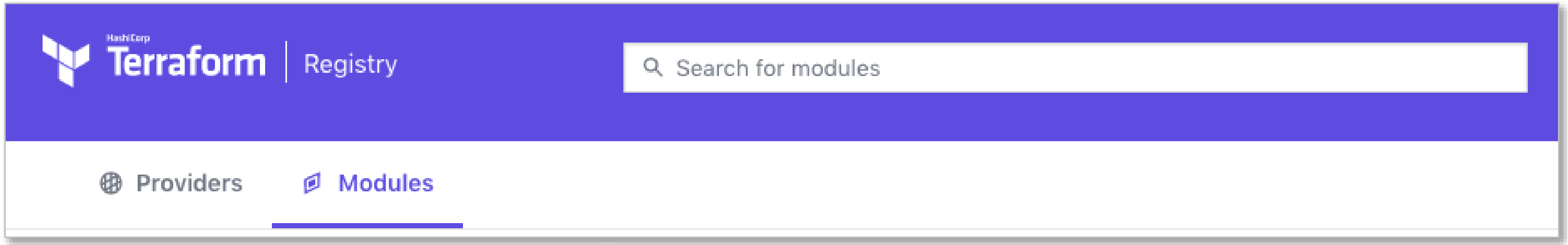
Module

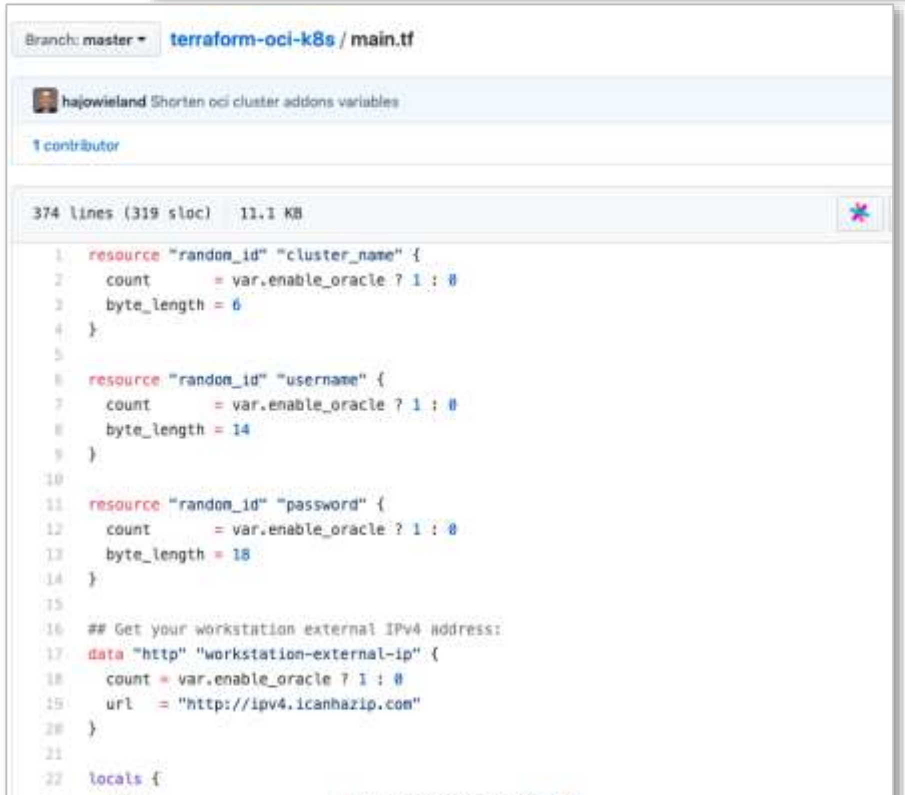


```
#   name_prefix    = var.name_prefix
#   freeform_tags = var.freeform_tags
# }

# Creates a compartment

module "compartment" {
  source         = "./modules/compartment"
  tenancy_ocid   = var.tenancy_ocid
  name_prefix    = var.name_prefix
  freeform_tags = var.freeform_tags
}
```

Terraform Core Concept

Module



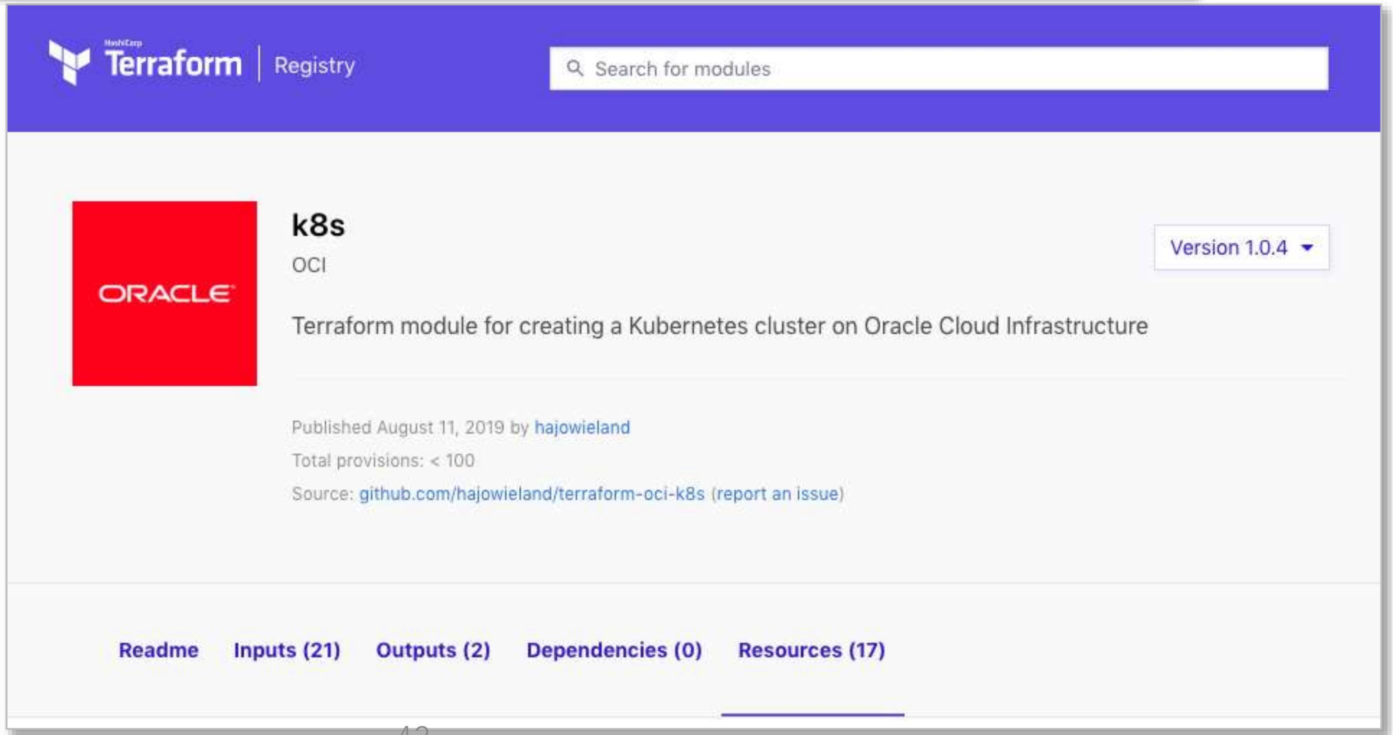

Branch: master ▾ terraform-oci-k8s / main.tf

hajowieland Shorten oci cluster addons variables

1 contributor

374 lines (319 sloc) 11.1 KB

```
resource "random_id" "cluster_name" {
  count       = var.enable_oracle ? 1 : 0
  byte_length = 6
}

resource "random_id" "username" {
  count       = var.enable_oracle ? 1 : 0
  byte_length = 14
}

resource "random_id" "password" {
  count       = var.enable_oracle ? 1 : 0
  byte_length = 18
}

## Get your workstation external IPv4 address:
data "http" "workstation-external-ip" {
  count = var.enable_oracle ? 1 : 0
  url   = "http://ipv4.icanhazip.com"
}

locals {
```



k8s

OCI

Terraform module for creating a Kubernetes cluster on Oracle Cloud Infrastructure

Published August 11, 2019 by hajowieland

Total provisions: < 100

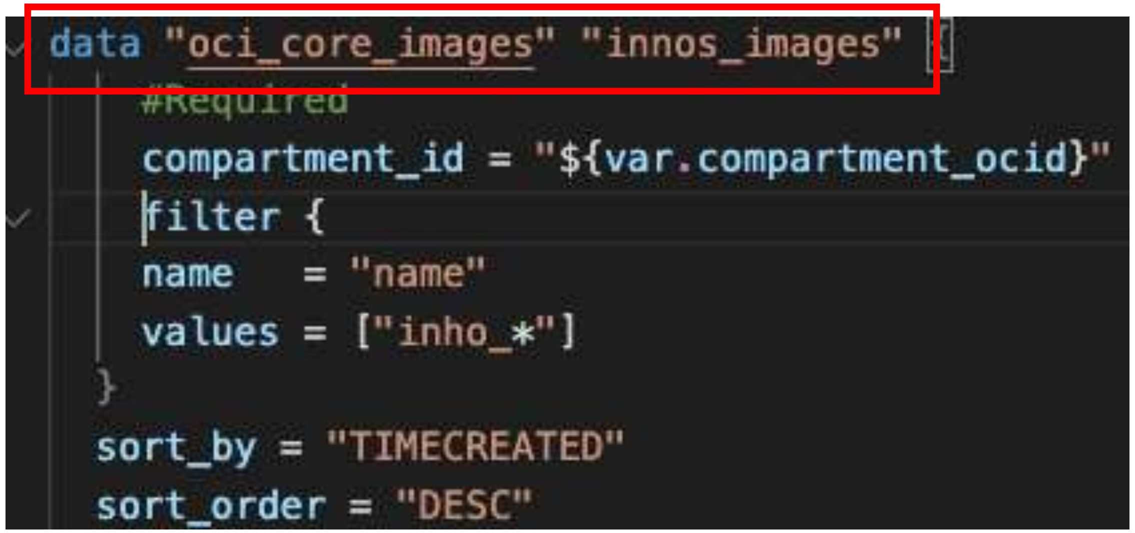
Source: github.com/hajowieland/terraform-oci-k8s (report an issue)

Readme Inputs (21) Outputs (2) Dependencies (0) Resources (17)

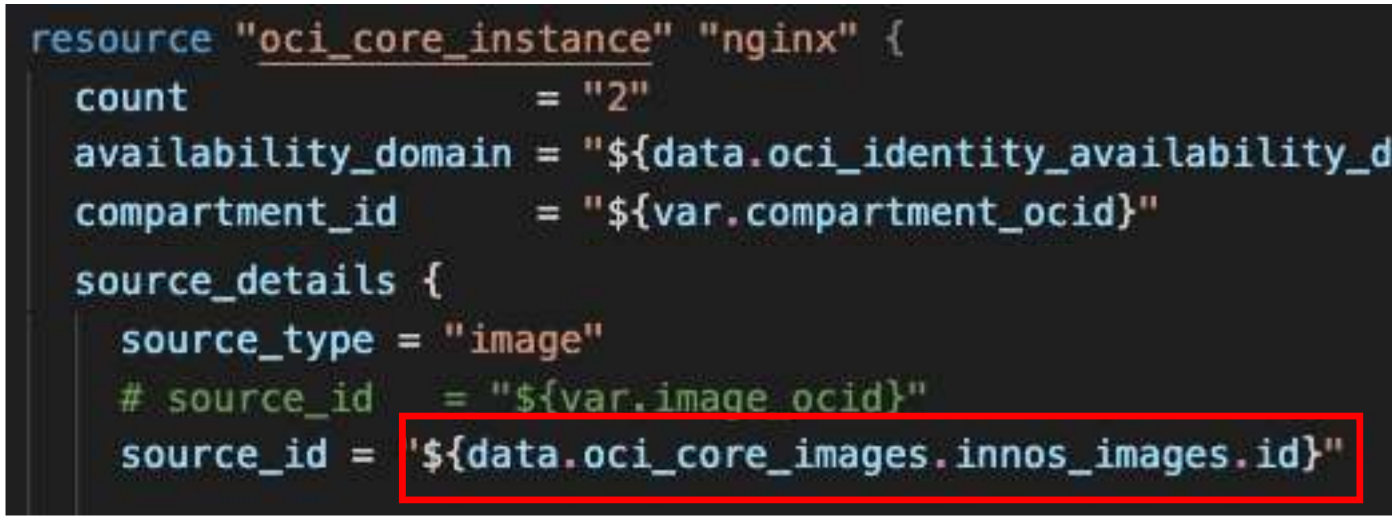
Terraform Core Concept #5

Data Source

- Data Source는 읽기 전용의 설정 정보를 담을 수 있는 객체로 예를 들면 Provider의 정보를 조회해서 읽기 전용 정보로 제공할 수 있다.
  ex) 새로 생성된 VM Based Image ID, 기존 LB의 ID등
- variable은 데이터를 자유롭게 설정하는데 반해 Data Source는 읽기 전용이다.



```
data "oci_core_images" "innos_images" {
    #Required
    compartment_id = "${var.compartment_ocid}"
    filter {
    name    = "name"
    values = ["inho_*"]
  }
  sort_by = "TIMECREATED"
  sort_order = "DESC"
```

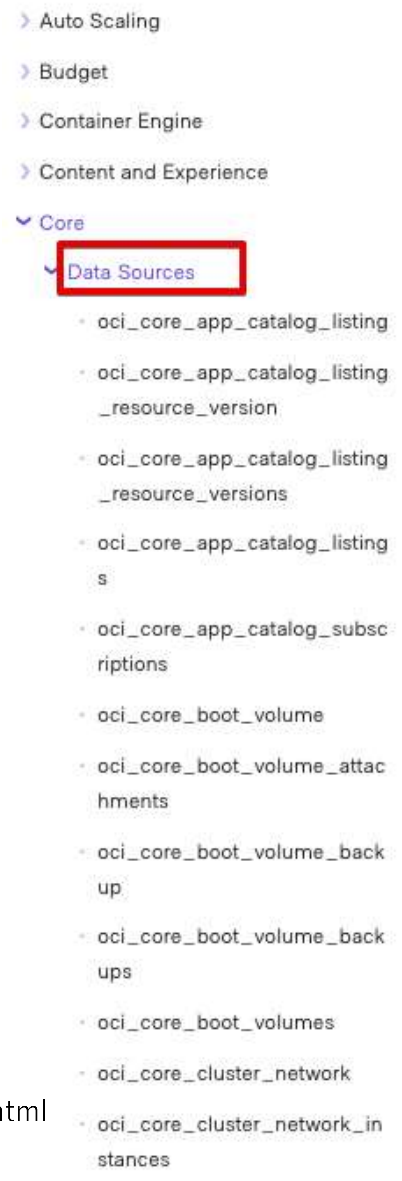


```
resource "oci_core_instance" "nginx" {
  count               = "2"
  availability_domain = "${data.oci_identity_availability_d
  compartment_id      = "${var.compartment_ocid}"
  source_details {
    source_type = "image"
    # source_id   = "${var.image_ocid}"
    source_id = "${data.oci_core_images.innos_images.id}"
```

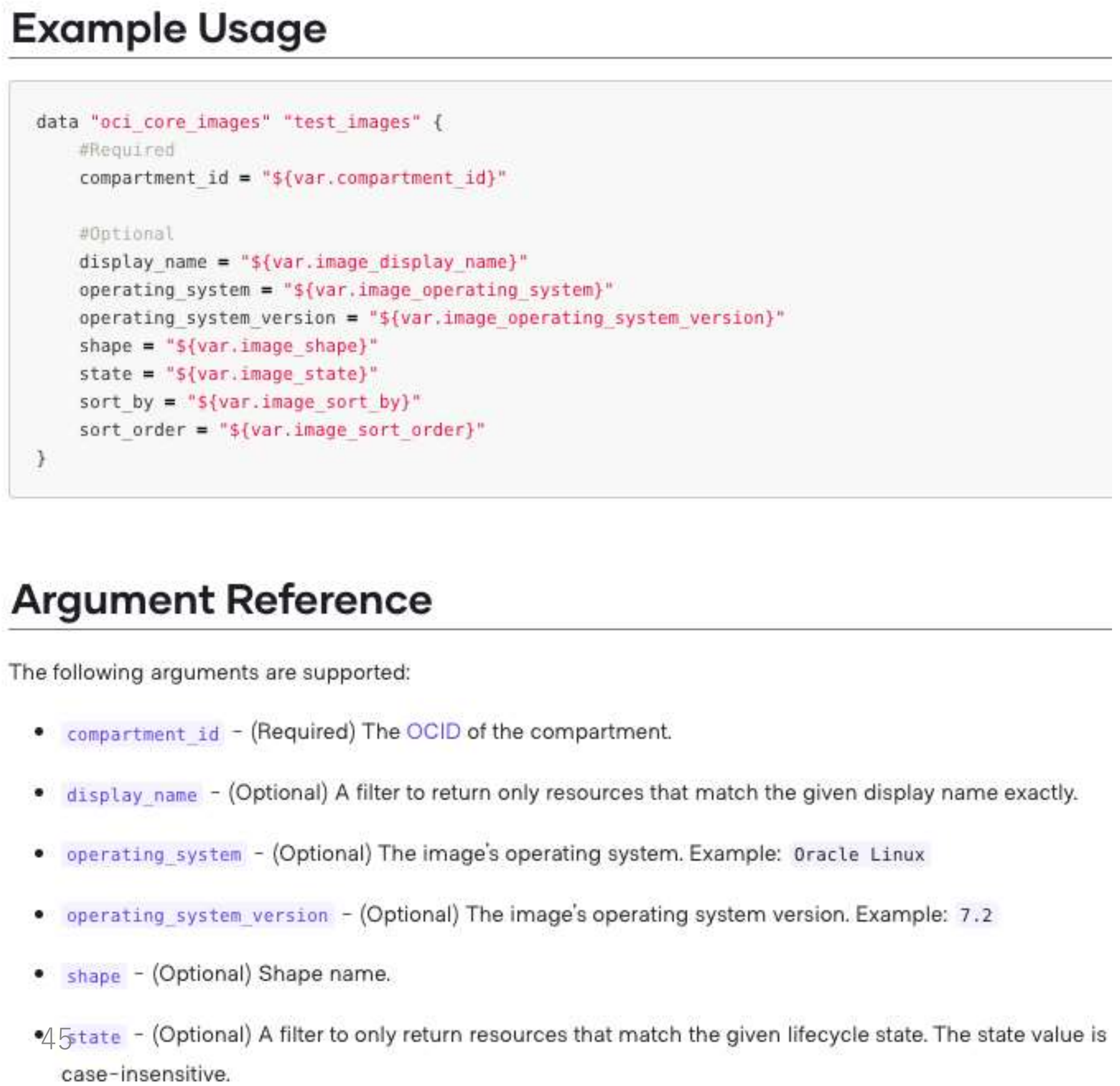
Terraform Core Concept #5

Data Source

- Data Source는 읽기 전용의 설정 정보를 담을 수 있는 객체로 Provider의 정보를 조회해서 접근할 수 있다.
- ex) 새로 생성된 VM Based Image ID, 기존 LB의 ID등
- variable은 데이터를 자유롭게 설정하는데 반해 Data Source는 읽기 전용이다.



- Auto Scaling
- Budget
- Container Engine
- Content and Experience
- Core
  - Data Sources
    - oci_core_app_catalog_listing
    - oci_core_app_catalog_listing_resource_version
    - oci_core_app_catalog_listing_resource_versions
    - oci_core_app_catalog_listings
    - oci_core_app_catalog_subscriptions
    - oci_core_boot_volume
    - oci_core_boot_volume_attachments
    - oci_core_boot_volume_backup
    - oci_core_boot_volume_backups
    - oci_core_boot_volumes
    - oci_core_cluster_network
    - oci_core_cluster_network_instances



Example Usage

```
data "oci_core_images" "test_images" {
    #Required
    compartment_id = "${var.compartment_id}"

    #Optional
    display_name = "${var.image_display_name}"
    operating_system = "${var.image_operating_system}"
    operating_system_version = "${var.image_operating_system_version}"
    shape = "${var.image_shape}"
    state = "${var.image_state}"
    sort_by = "${var.image_sort_by}"
    sort_order = "${var.image_sort_order}"
}
```

Argument Reference

The following arguments are supported:

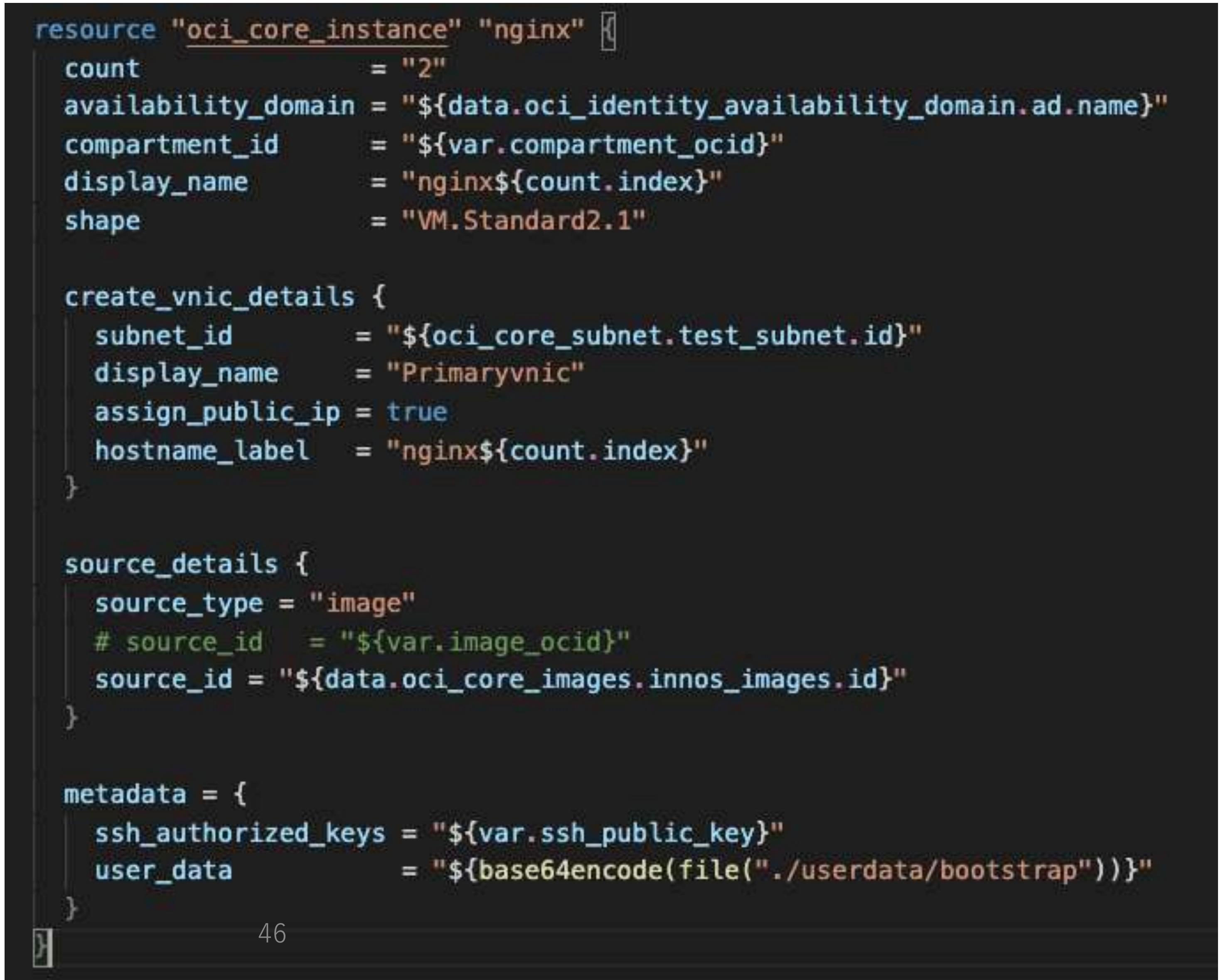
- `compartment_id` - (Required) The OCID of the compartment.
- `display_name` - (Optional) A filter to return only resources that match the given display name exactly.
- `operating_system` - (Optional) The image's operating system. Example: `Oracle Linux`
- `operating_system_version` - (Optional) The image's operating system version. Example: `7.2`
- `shape` - (Optional) Shape name.
- `state` - (Optional) A filter to only return resources that match the given lifecycle state. The state value is case-insensitive.

https://www.terraform.io/docs/providers/oci/d/core_images.html

Terraform Core Concept #6

Dependency

```
Project
├── compute.tf
├── vcn.tf
├── userdata
│   └── bootstrap
└── variables.tf
```

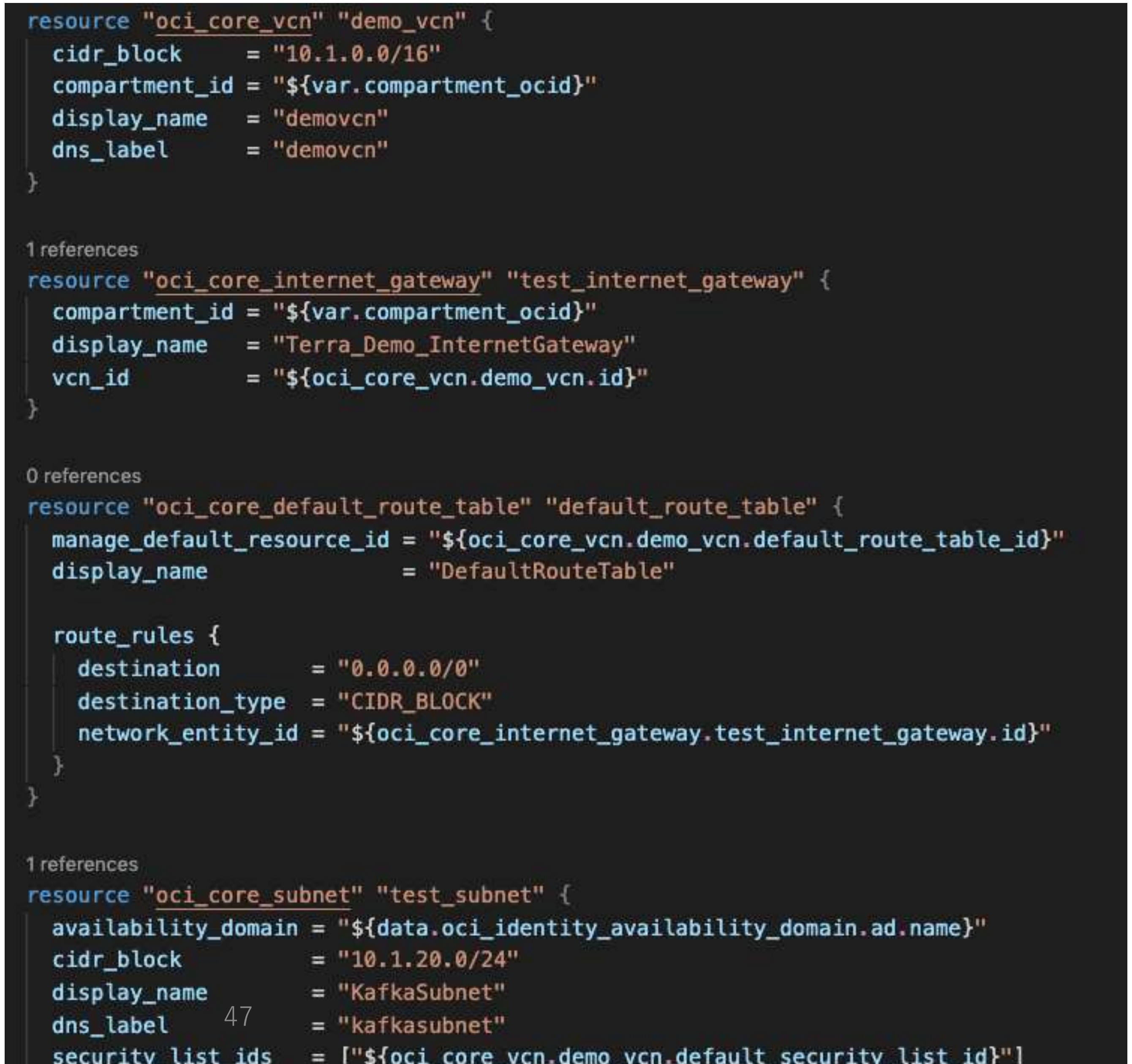


```
resource "oci_core_instance" "nginx" {
  count               = "2"
  availability_domain = "${data.oci_identity_availability_domain.ad.name}"
  compartment_id      = "${var.compartment_ocid}"
  display_name        = "nginx${count.index}"
  shape               = "VM.Standard2.1"

  create_vnic_details {
    subnet_id        = "${oci_core_subnet.test_subnet.id}"
    display_name     = "Primaryvnic"
    assign_public_ip = true
    hostname_label   = "nginx${count.index}"
  }

  source_details {
    source_type = "image"
    # source_id   = "${var.image_ocid}"
    source_id = "${data.oci_core_images.innos_images.id}"
  }

  metadata = {
    ssh_authorized_keys = "${var.ssh_public_key}"
    user_data           = "${base64encode(file("./userdata/bootstrap"))}"
  }
}
```

Terraform Core Concept #6

Dependency

```
Project
├──── compute.tf
├──── vcn.tf
├──── userdata
│    └──── bootstrap
└──── variables.tf
```

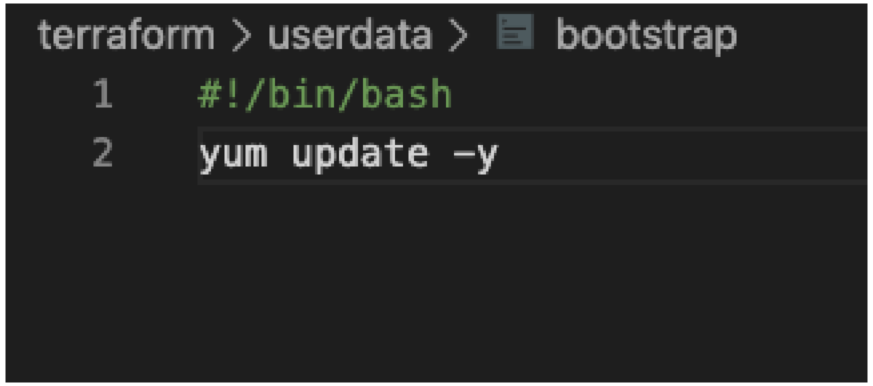


```
resource "oci_core_vcn" "demo_vcn" {
  cidr_block     = "10.1.0.0/16"
  compartment_id = "${var.compartment_ocid}"
  display_name   = "demovcn"
  dns_label      = "demovcn"
}

1 references
resource "oci_core_internet_gateway" "test_internet_gateway" {
  compartment_id = "${var.compartment_ocid}"
  display_name   = "Terra_Demo_InternetGateway"
  vcn_id         = "${oci_core_vcn.demo_vcn.id}"
}

0 references
resource "oci_core_default_route_table" "default_route_table" {
  manage_default_resource_id = "${oci_core_vcn.demo_vcn.default_route_table_id}"
  display_name               = "DefaultRouteTable"

  route_rules {
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    network_entity_id = "${oci_core_internet_gateway.test_internet_gateway.id}"
  }
}

1 references
resource "oci_core_subnet" "test_subnet" {
  availability_domain = "${data.oci_identity_availability_domain.ad.name}"
  cidr_block          = "10.1.20.0/24"
  display_name        = "KafkaSubnet"
  dns_label           = "kafkasubnet"
  security_list_ids   = ["${oci_core_vcn.demo_vcn.default_security_list_id}"]
```

Terraform Core Concept #6

Dependency

```
Project
├── compute.tf
├── vcn.tf
├── userdata
│   └── bootstrap
└── variables.tf
```



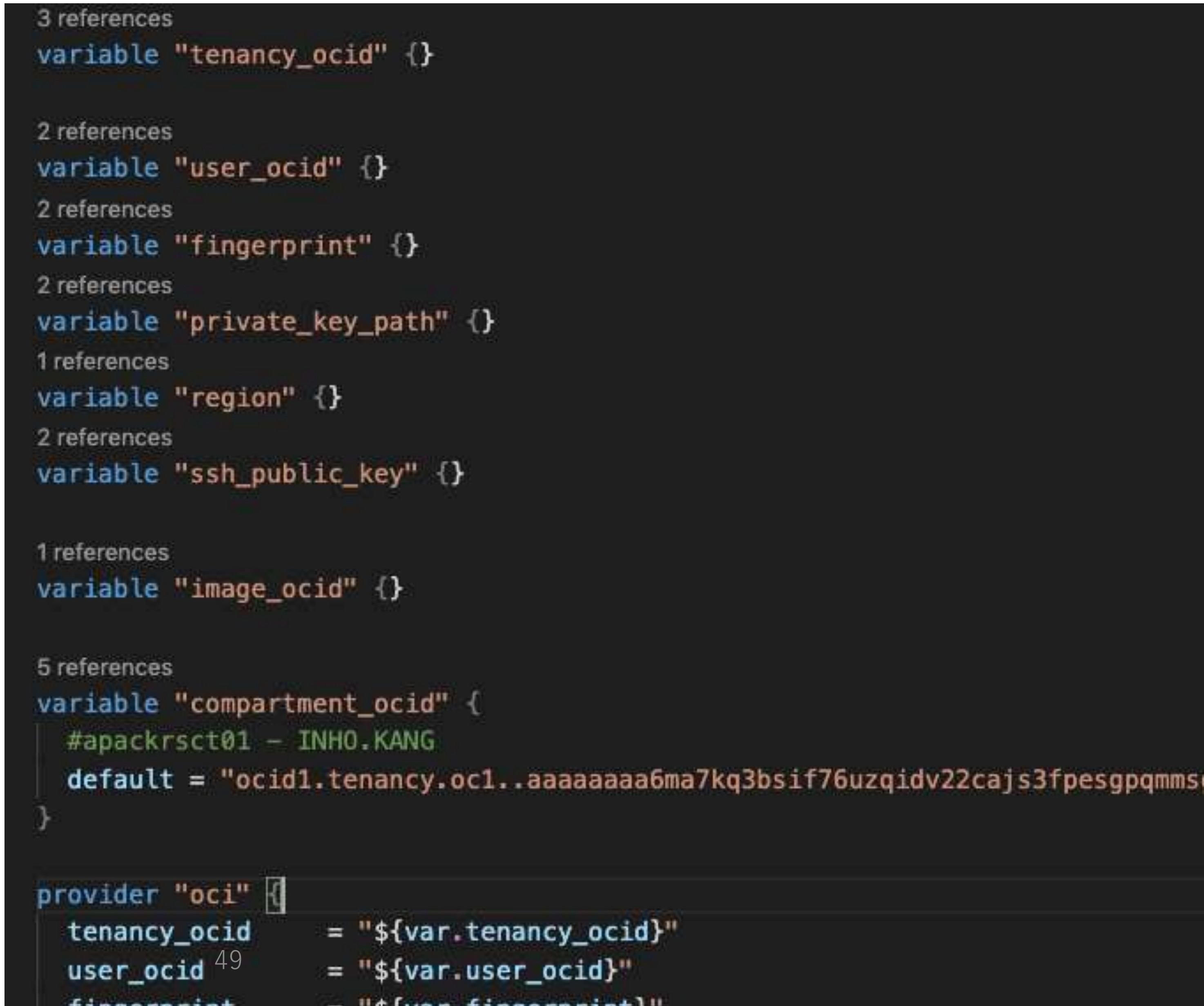
terraform > userdata > bootstrap

```
#!/bin/bash
yum update -y
```

Terraform Core Concept #6

Dependency

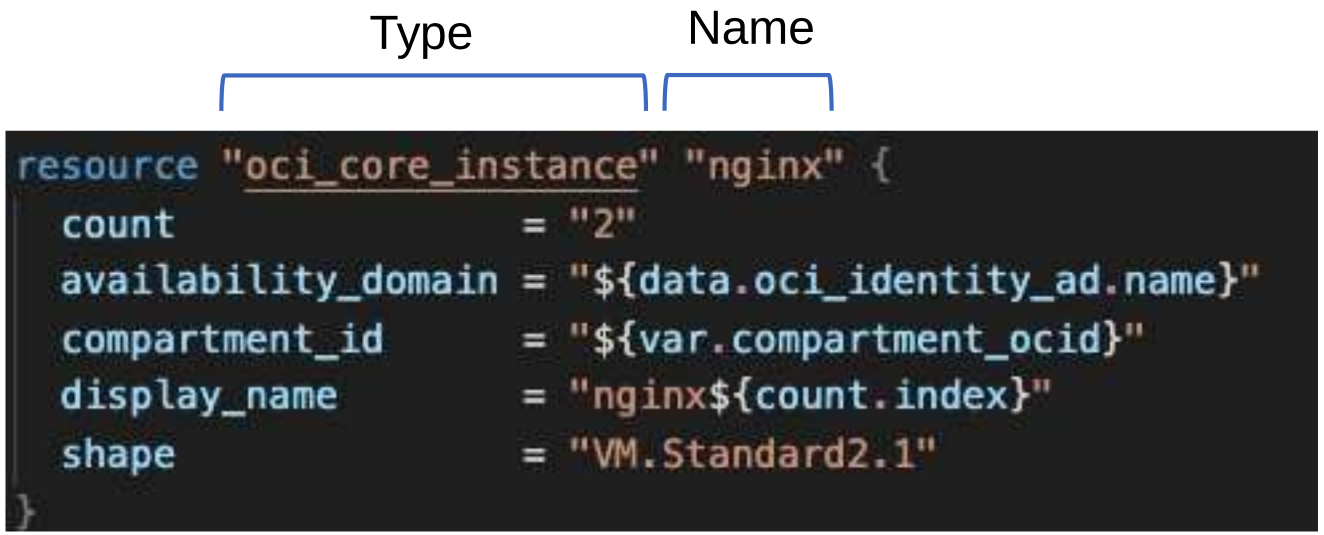
```
Project
├── compute.tf
├── vcn.tf
├── userdata
│   └── bootstrap
└── variables.tf
```



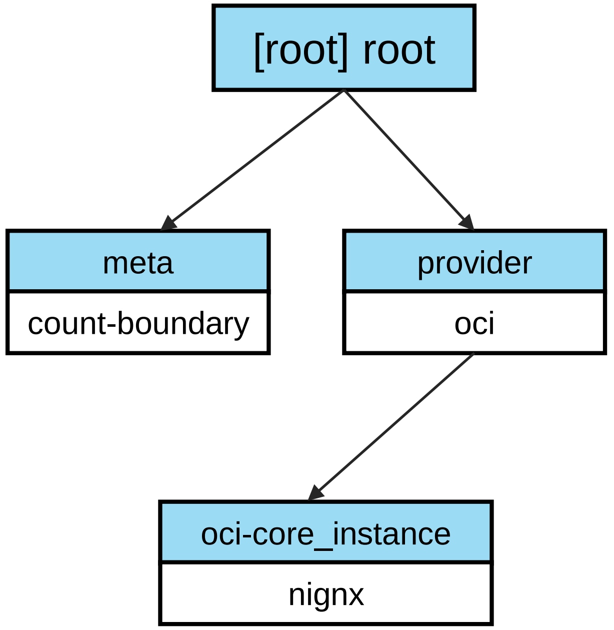
```
3 references
variable "tenancy_ocid" {}

2 references
variable "user_ocid" {}
2 references
variable "fingerprint" {}
2 references
variable "private_key_path" {}
1 references
variable "region" {}
2 references
variable "ssh_public_key" {}

1 references
variable "image_ocid" {}

5 references
variable "compartment_ocid" {
  #apackrsct01 - INHO.KANG
  default = "ocid1.tenancy.oc1..aaaaaaaa6ma7kq3bsif76uzqidv22cajs3fpesgpqmms
}

provider "oci" {
  tenancy_ocid     = "${var.tenancy_ocid}"
  user_ocid        = "${var.user_ocid}"
```

Terraform Core Concept #6

Dependency



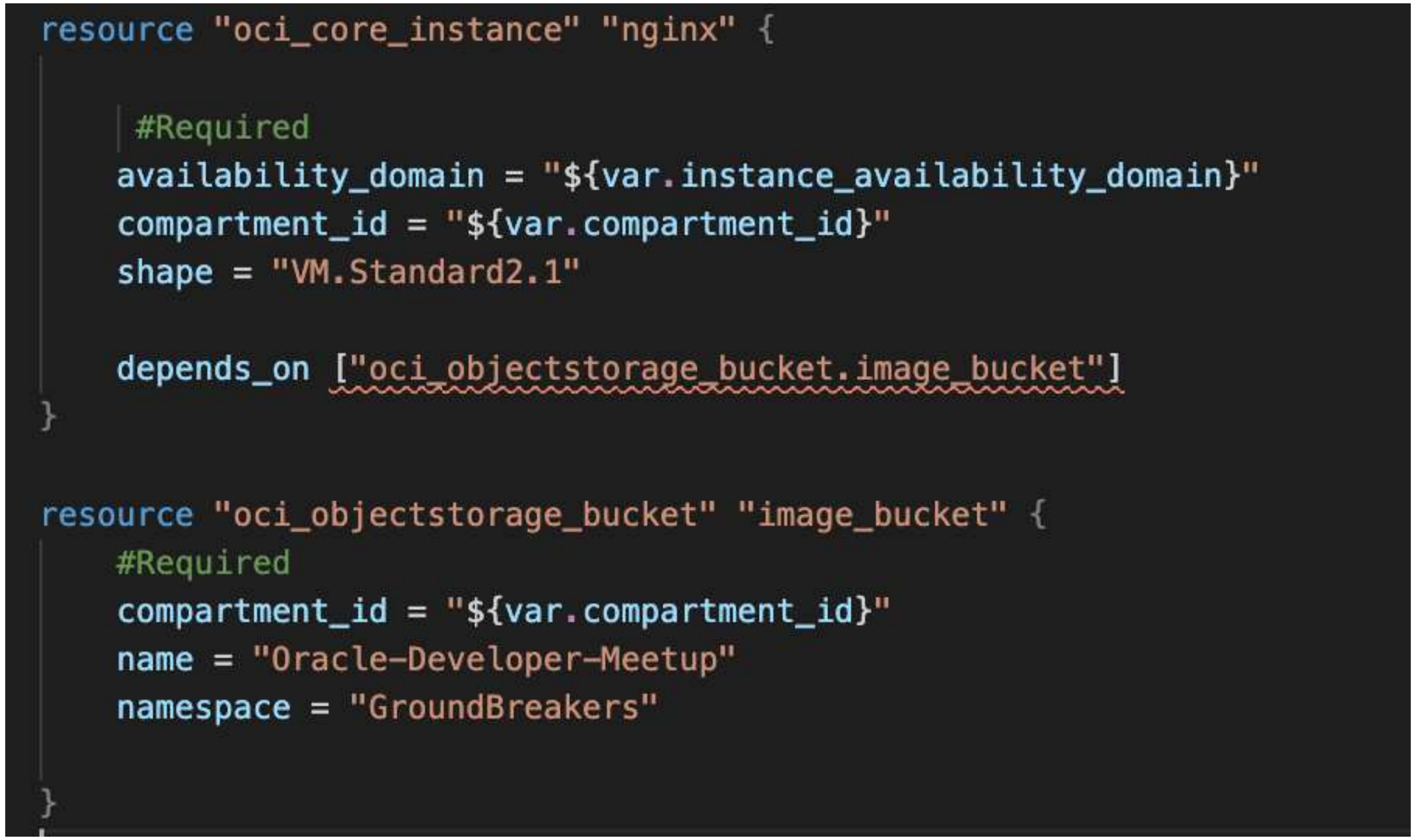
```
resource "oci_core_instance" "nginx" {
  count               = "2"
  availability_domain = "${data.oci_identity_ad.name}"
  compartment_id      = "${var.compartment_ocid}"
  display_name        = "nginx${count.index}"
  shape               = "VM.Standard2.1"
}
```



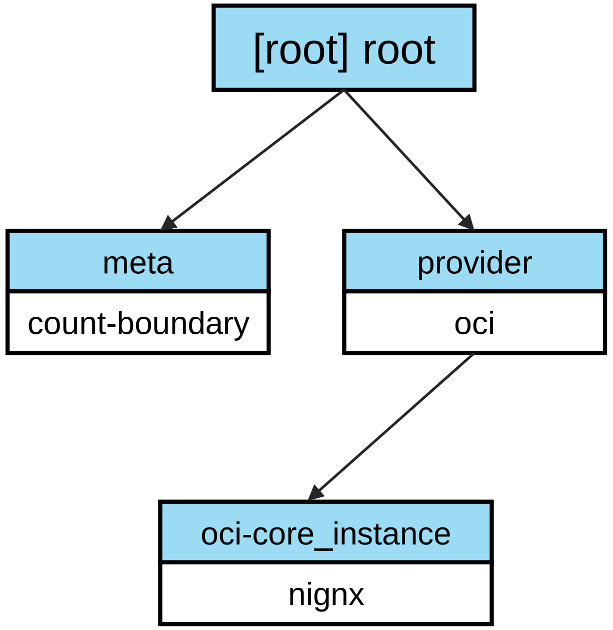
Terraform Core Concept #6

Dependency

- depends_on



```
resource "oci_core_instance" "nginx" {

    #Required
    availability_domain = "${var.instance_availability_domain}"
    compartment_id = "${var.compartment_id}"
    shape = "VM.Standard2.1"

    depends_on ["oci_objectstorage_bucket.image_bucket"]
}

resource "oci_objectstorage_bucket" "image_bucket" {
    #Required
    compartment_id = "${var.compartment_id}"
    name = "Oracle-Developer-Meetup"
    namespace = "GroundBreakers"

}
```

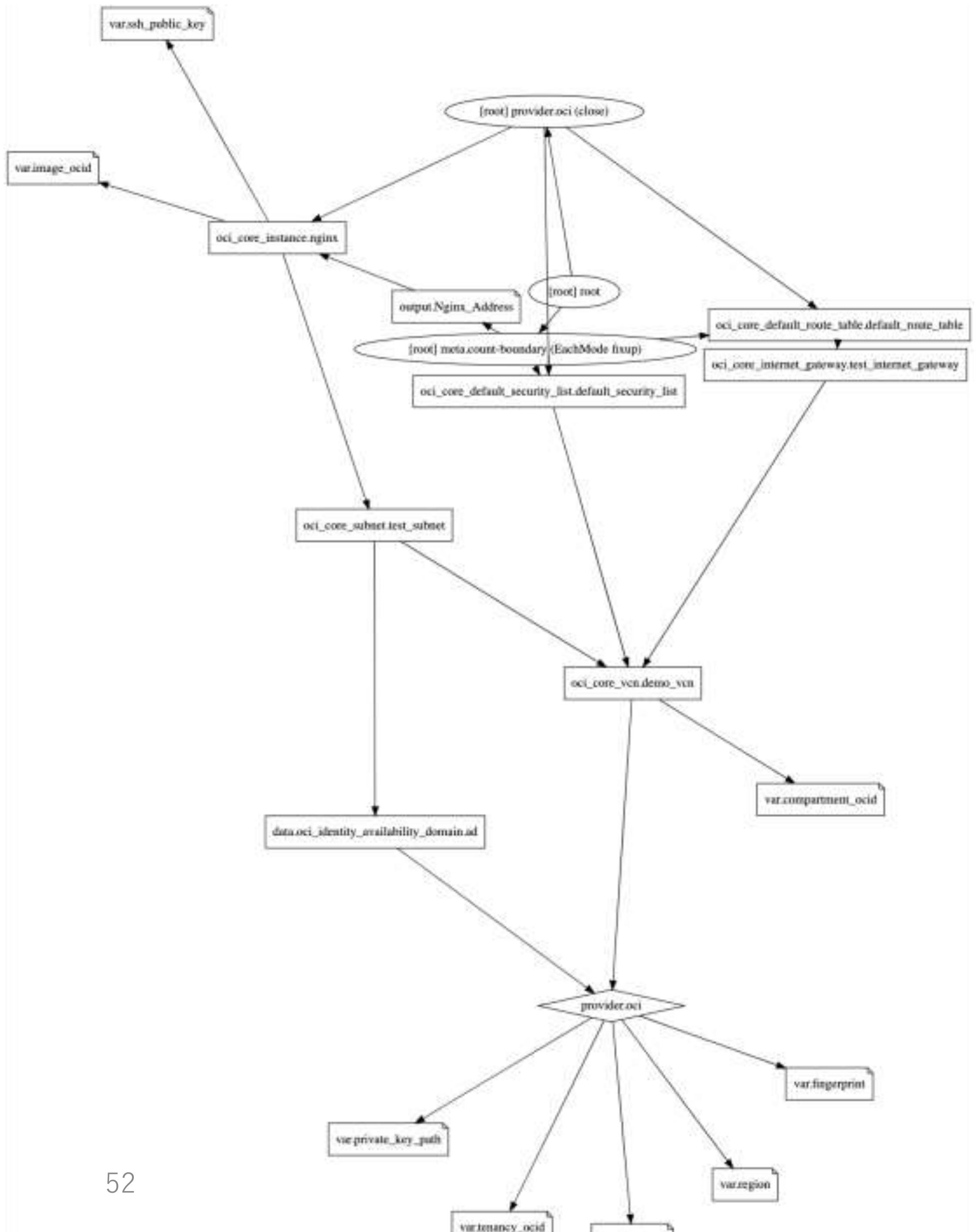


Terraform Core Concept #6

Dependency

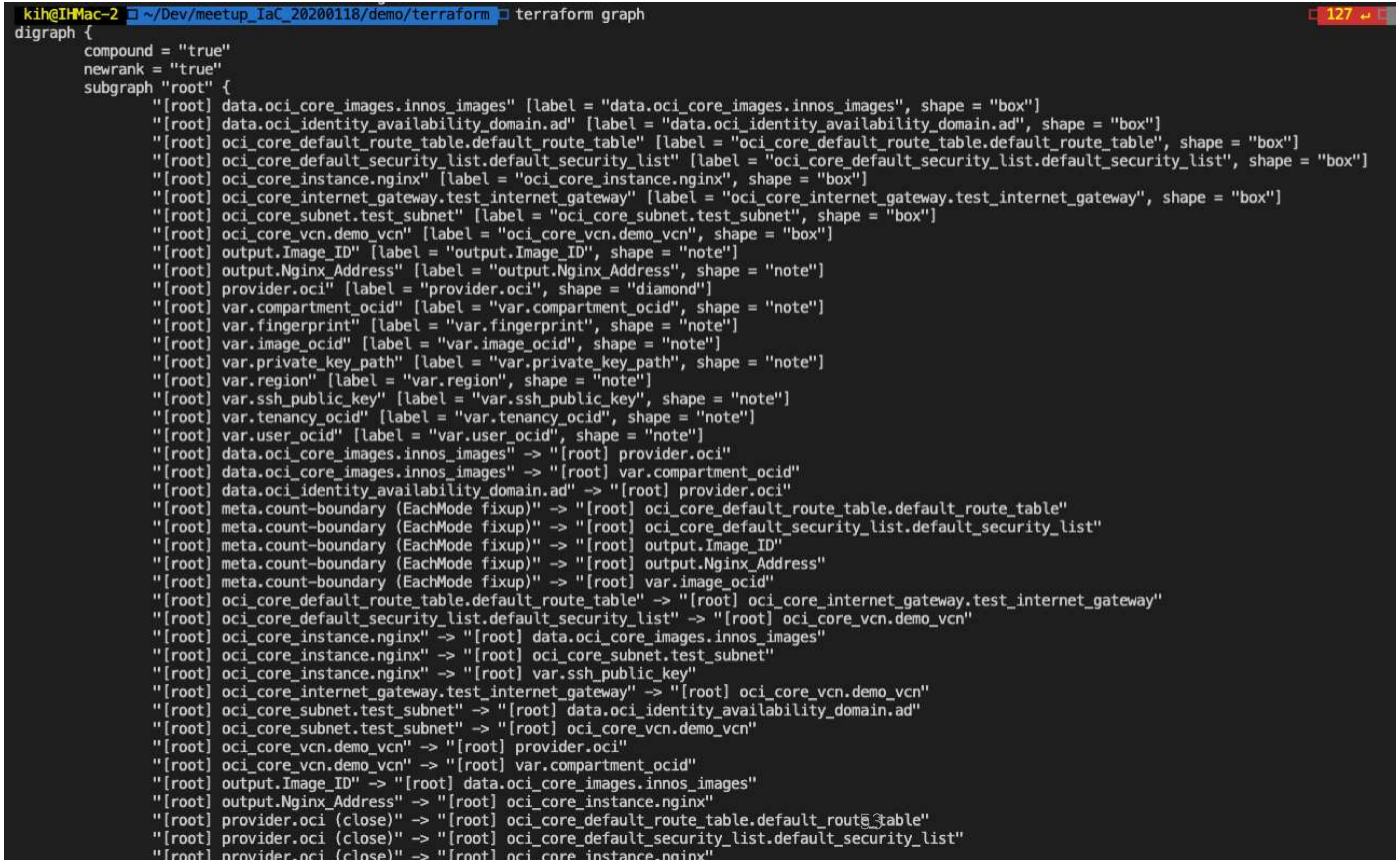
```
Project
├── compute.tf
├── vcn.tf
├── userdata
│   └── bootstrap
└── variables.tf
```

http://www.webgraphviz.com/



var.ssh_public_key
[root] provider.oci (close)
var.image_ocid
oci_core_instance.nginx
output.Nginx_Address
[root] root
oci_core_default_route_table.default_route_table
[root] meta.count-boundary (EachMode fixup)
oci_core_internet_gateway.test_internet_gateway
oci_core_default_security_list.default_security_list
oci_core_subnet.test_subnet
oci_core_vcn.demo_vcn
var.compartment_ocid
data.oci_identity_availability_domain.ad
provider.oci
var.fingerprint
var.private_key_path
var.region
var.tenancy_ocid

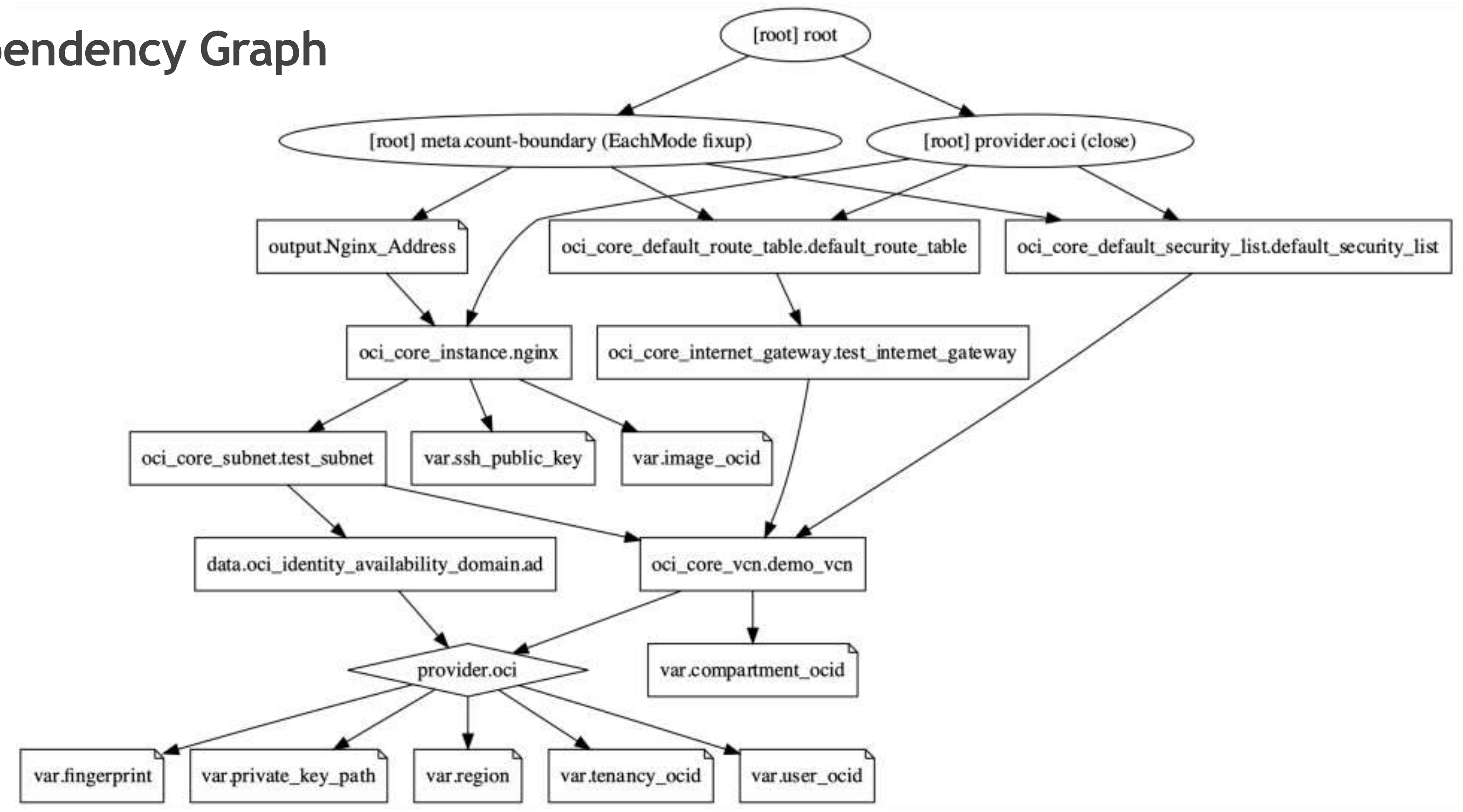
Terraform Core Concept #6



```
kih@IHMac-2  ~/Dev/meetup_IaC_20200118/demo/terraform  terraform graph                                    127 ↵
digraph {
        compound = "true"
        newrank = "true"
        subgraph "root" {
                "[root] data.oci_core_images.innos_images" [label = "data.oci_core_images.innos_images", shape = "box"]
                "[root] data.oci_identity_availability_domain.ad" [label = "data.oci_identity_availability_domain.ad", shape = "box"]
                "[root] oci_core_default_route_table.default_route_table" [label = "oci_core_default_route_table.default_route_table", shape = "box"]
                "[root] oci_core_default_security_list.default_security_list" [label = "oci_core_default_security_list.default_security_list", shape = "box"]
                "[root] oci_core_instance.nginx" [label = "oci_core_instance.nginx", shape = "box"]
                "[root] oci_core_internet_gateway.test_internet_gateway" [label = "oci_core_internet_gateway.test_internet_gateway", shape = "box"]
                "[root] oci_core_subnet.test_subnet" [label = "oci_core_subnet.test_subnet", shape = "box"]
                "[root] oci_core_vcn.demo_vcn" [label = "oci_core_vcn.demo_vcn", shape = "box"]
                "[root] output.Image_ID" [label = "output.Image_ID", shape = "note"]
                "[root] output.Nginx_Address" [label = "output.Nginx_Address", shape = "note"]
                "[root] provider.oci" [label = "provider.oci", shape = "diamond"]
                "[root] var.compartment_ocid" [label = "var.compartment_ocid", shape = "note"]
                "[root] var.fingerprint" [label = "var.fingerprint", shape = "note"]
                "[root] var.image_ocid" [label = "var.image_ocid", shape = "note"]
                "[root] var.private_key_path" [label = "var.private_key_path", shape = "note"]
                "[root] var.region" [label = "var.region", shape = "note"]
                "[root] var.ssh_public_key" [label = "var.ssh_public_key", shape = "note"]
                "[root] var.tenancy_ocid" [label = "var.tenancy_ocid", shape = "note"]
                "[root] var.user_ocid" [label = "var.user_ocid", shape = "note"]
                "[root] data.oci_core_images.innos_images" -> "[root] provider.oci"
                "[root] data.oci_core_images.innos_images" -> "[root] var.compartment_ocid"
                "[root] data.oci_identity_availability_domain.ad" -> "[root] provider.oci"
                "[root] meta.count-boundary (EachMode fixup)" -> "[root] oci_core_default_route_table.default_route_table"
                "[root] meta.count-boundary (EachMode fixup)" -> "[root] oci_core_default_security_list.default_security_list"
                "[root] meta.count-boundary (EachMode fixup)" -> "[root] output.Image_ID"
                "[root] meta.count-boundary (EachMode fixup)" -> "[root] output.Nginx_Address"
                "[root] meta.count-boundary (EachMode fixup)" -> "[root] var.image_ocid"
                "[root] oci_core_default_route_table.default_route_table" -> "[root] oci_core_internet_gateway.test_internet_gateway"
                "[root] oci_core_default_security_list.default_security_list" -> "[root] oci_core_vcn.demo_vcn"
                "[root] oci_core_instance.nginx" -> "[root] data.oci_core_images.innos_images"
                "[root] oci_core_instance.nginx" -> "[root] oci_core_subnet.test_subnet"
                "[root] oci_core_instance.nginx" -> "[root] var.ssh_public_key"
                "[root] oci_core_internet_gateway.test_internet_gateway" -> "[root] oci_core_vcn.demo_vcn"
                "[root] oci_core_subnet.test_subnet" -> "[root] data.oci_identity_availability_domain.ad"
                "[root] oci_core_subnet.test_subnet" -> "[root] oci_core_vcn.demo_vcn"
                "[root] oci_core_vcn.demo_vcn" -> "[root] provider.oci"
                "[root] oci_core_vcn.demo_vcn" -> "[root] var.compartment_ocid"
                "[root] output.Image_ID" -> "[root] data.oci_core_images.innos_images"
                "[root] output.Nginx_Address" -> "[root] oci_core_instance.nginx"
                "[root] provider.oci (close)" -> "[root] oci_core_default_route_table.default_route_table"
                "[root] provider.oci (close)" -> "[root] oci_core_default_security_list.default_security_list"
                "[root] provider.oci (close)" -> "[root] oci_core_instance.nginx"
```

Terraform Core Concept #6

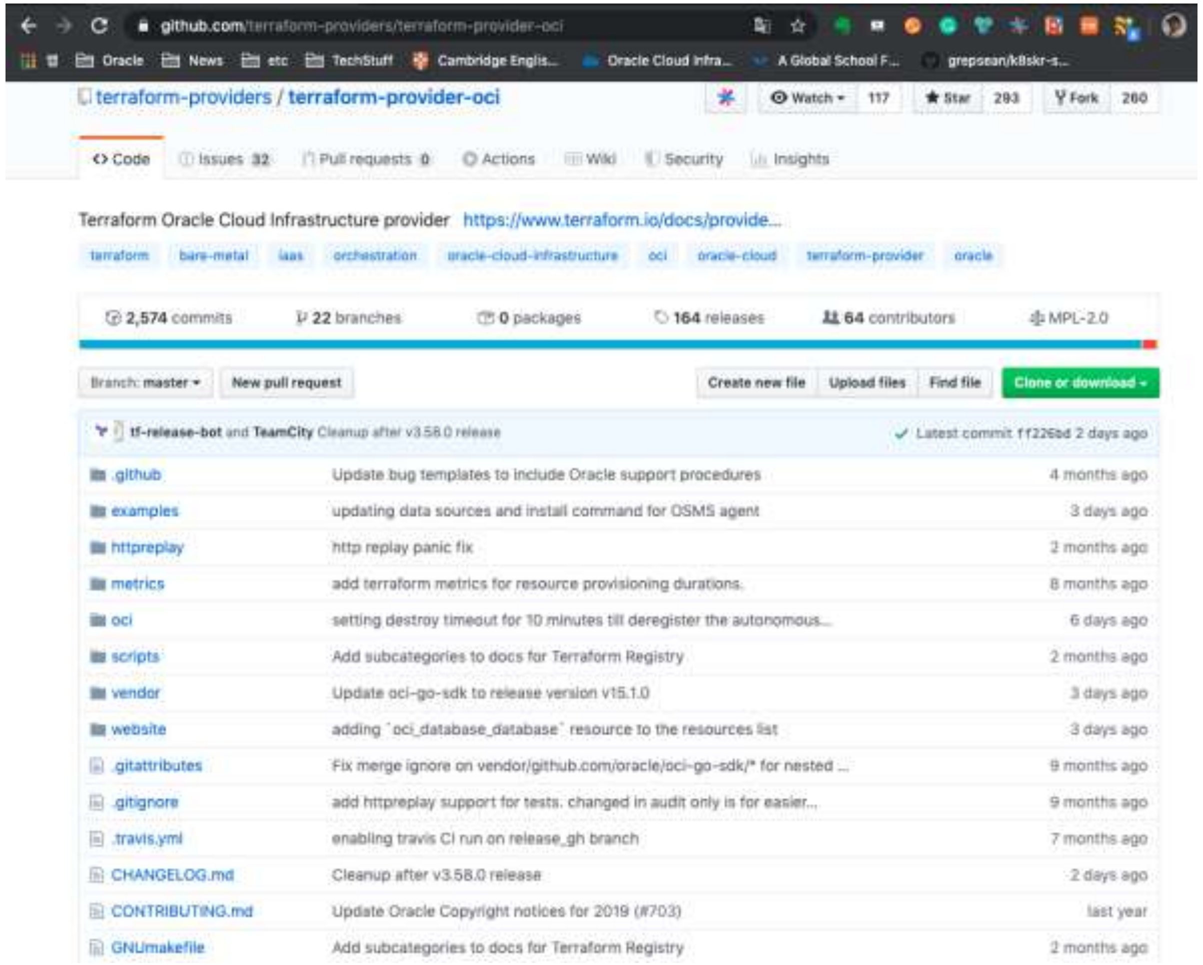
Dependency Graph


[root] root
[root] meta.count-boundary (EachMode fixup)
[root] provider.oci (close)
output.Nginx_Address
oci_core_default_route_table.default_route_table
oci_core_default_security_list.default_security_list
oci_core_instance.nginx
oci_core_internet_gateway.test_internet_gateway
oci_core_subnet.test_subnet
var.ssh_public_key
var.image_ocid
data.oci_identity_availability_domain.ad
oci_core_vcn.demo_vcn
provider.oci
var.compartment_ocid
var.fingerprint
var.private_key_path
var.region
var.tenancy_ocid
var.user_ocid

http://www.webgraphviz.com/ http://dreampuf.github.io/GraphvizOnline/

Terraform 실전

OCI Reference



terraform-providers / terraform-provider-oci

Terraform Oracle Cloud Infrastructure provider https://www.terraform.io/docs/provide...

terraform · bare-metal · iaas · orchestration · oracle-cloud-infrastructure · oci · oracle-cloud · terraform-provider · oracle

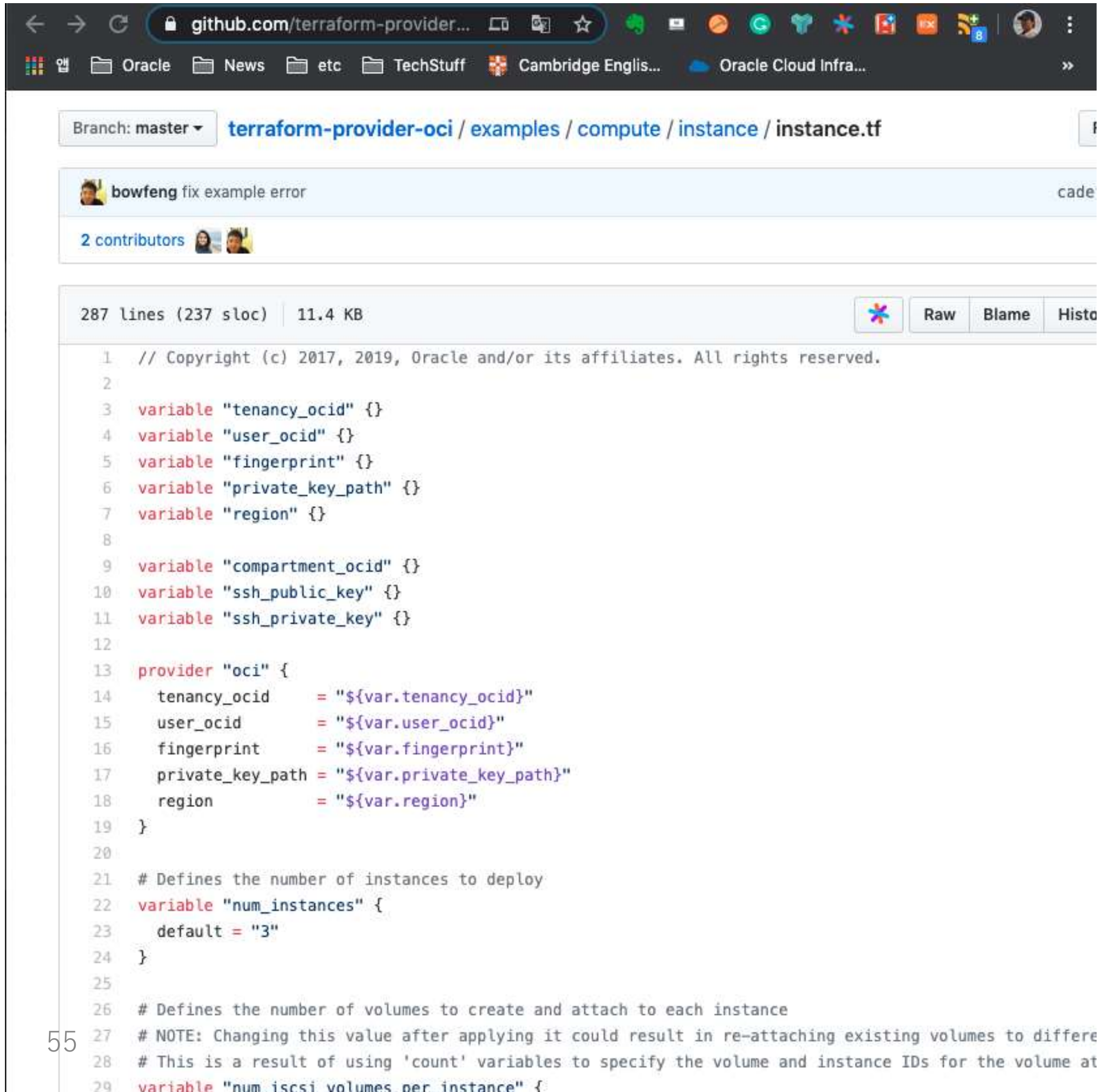
2,574 commits · 22 branches · 0 packages · 164 releases · 64 contributors · MPL-2.0

tf-release-bot and TeamCity Cleanup after v3.58.0 release — Latest commit ff226bd 2 days ago

.github	Update bug templates to include Oracle support procedures	4 months ago
examples	updating data sources and install command for OSMS agent	3 days ago
httpreplay	http replay panic fix	2 months ago
metrics	add terraform metrics for resource provisioning durations.	8 months ago
oci	setting destroy timeout for 10 minutes till deregister the autonomous...	6 days ago
scripts	Add subcategories to docs for Terraform Registry	2 months ago
vendor	Update oci-go-sdk to release version v15.1.0	3 days ago
website	adding `oci_database_database` resource to the resources list	3 days ago
.gitattributes	Fix merge ignore on vendor/github.com/oracle/oci-go-sdk/* for nested ...	9 months ago
.gitignore	add httpreplay support for tests. changed in audit only is for easier...	9 months ago
.travis.yml	enabling travis CI run on release_gh branch	7 months ago
CHANGELOG.md	Cleanup after v3.58.0 release	2 days ago
CONTRIBUTING.md	Update Oracle Copyright notices for 2019 (#703)	last year
GNUmakefile	Add subcategories to docs for Terraform Registry	2 months ago

https://github.com/terraform-providers/terraform-provider-oci

https://github.com/terraform-providers/terraform-provider-oci/blob/master/examples/compute/instance/instance.tf



terraform-provider-oci / examples / compute / instance / instance.tf

bowfeng fix example error

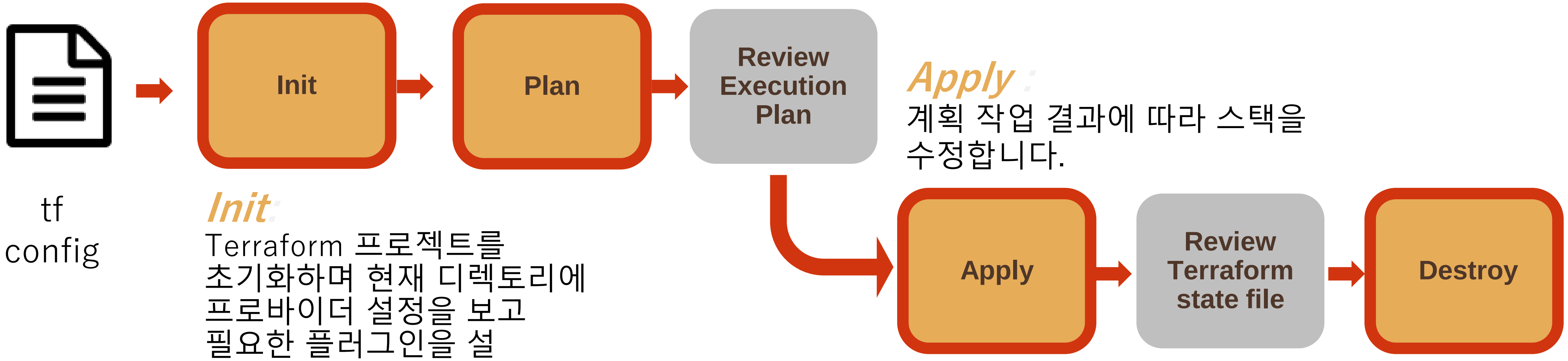
2 contributors

287 lines (237 sloc) 11.4 KB

```
// Copyright (c) 2017, 2019, Oracle and/or its affiliates. All rights reserved.

variable "tenancy_ocid" {}
variable "user_ocid" {}
variable "fingerprint" {}
variable "private_key_path" {}
variable "region" {}

variable "compartment_ocid" {}
variable "ssh_public_key" {}
variable "ssh_private_key" {}

provider "oci" {
  tenancy_ocid     = "${var.tenancy_ocid}"
  user_ocid        = "${var.user_ocid}"
  fingerprint      = "${var.fingerprint}"
  private_key_path = "${var.private_key_path}"
  region           = "${var.region}"
}

# Defines the number of instances to deploy
variable "num_instances" {
  default = "3"
}

# Defines the number of volumes to create and attach to each instance
# NOTE: Changing this value after applying it could result in re-attaching existing volumes to differe
# This is a result of using 'count' variables to specify the volume and instance IDs for the volume at
variable "num_iscsi_volumes_per_instance" {
```

Terraform Workflow

Plan :
Terraform을 구문 분석하고
영향을 받는 OCI 리소스에
대한 실행 계획을 반환



Apply :
계획 작업 결과에 따라 스택을
수정합니다.

Init:
Terraform 프로젝트를
초기화하며 현재 디렉토리에
프로바이더 설정을 보고
필요한 플러그인을 설

Destroy :
스택의 모든 리소스를
프로비저닝 해제

Terraform HCL 0.12

0.12 Enhancement

- Better error messages
- Reliable JSON Syntax - 1:1 mapping to Json
  - Comments in JSON
- Template Syntax Improvements
- Rich and Complex Value Types
  - Return Module resources as Object values
  - Maps of Maps? It’s possible!
- Conditionally Omitted Arguments
- Conditional Operator Improvements
- Splat Operator
- For and For-Each - Finally! - For nested blocks!

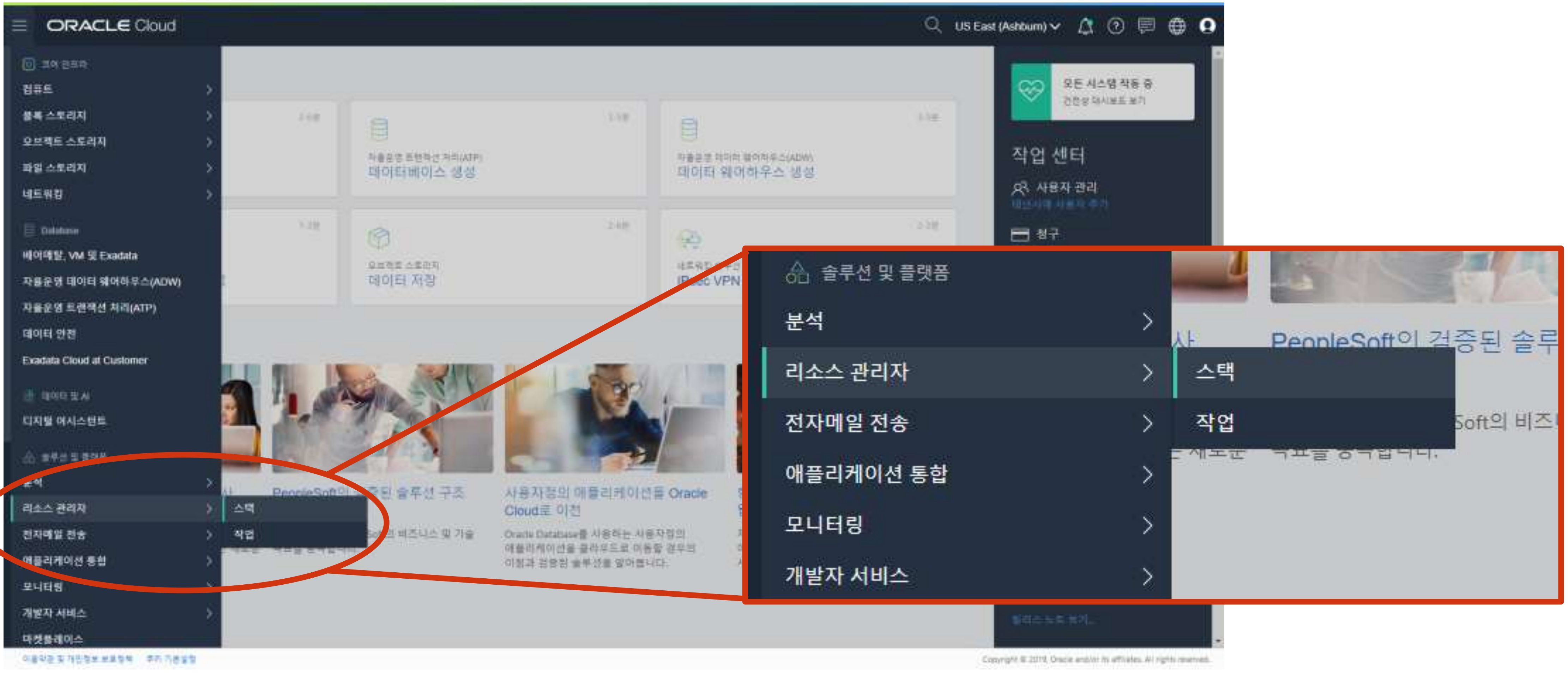
LIVE DEMO

- terraform graph

http://www.webgraphviz.com/

58

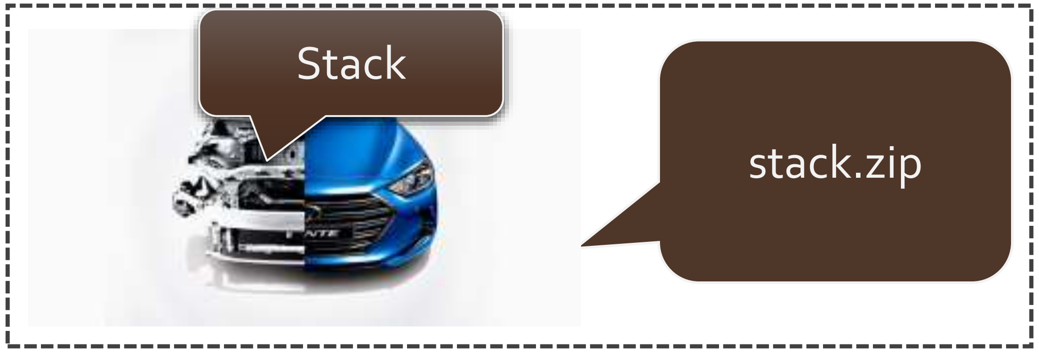
Resource Manager



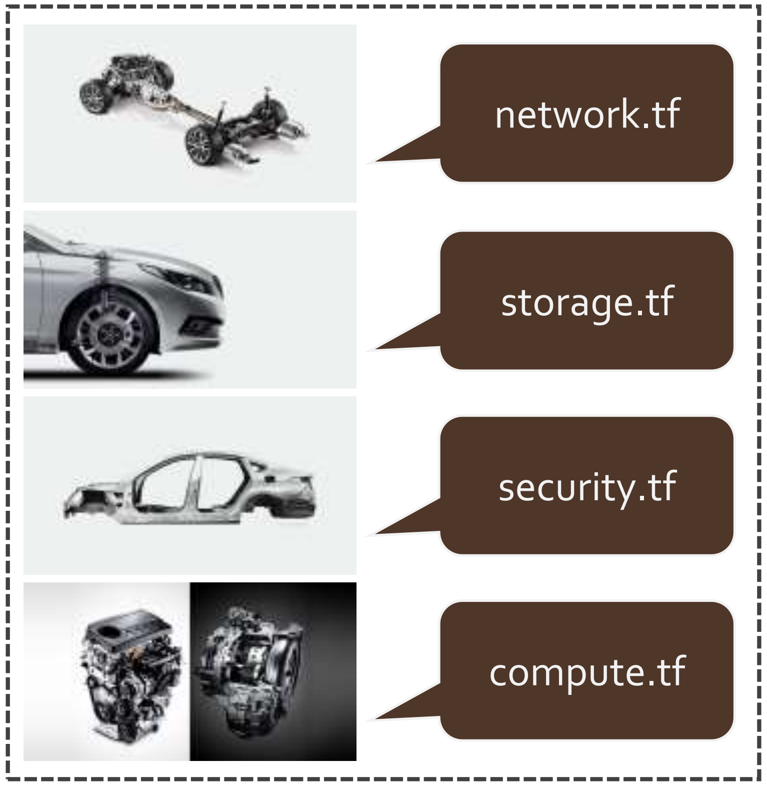
Resource Manager

Stack

zip file

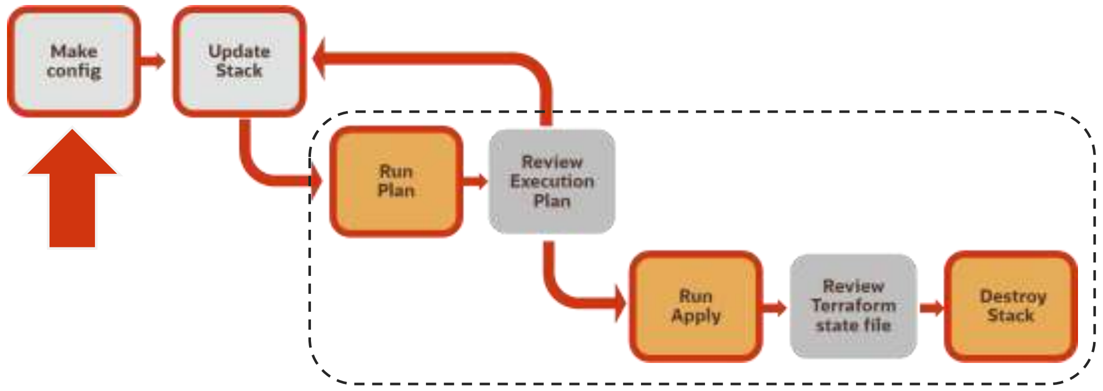


Terraform file

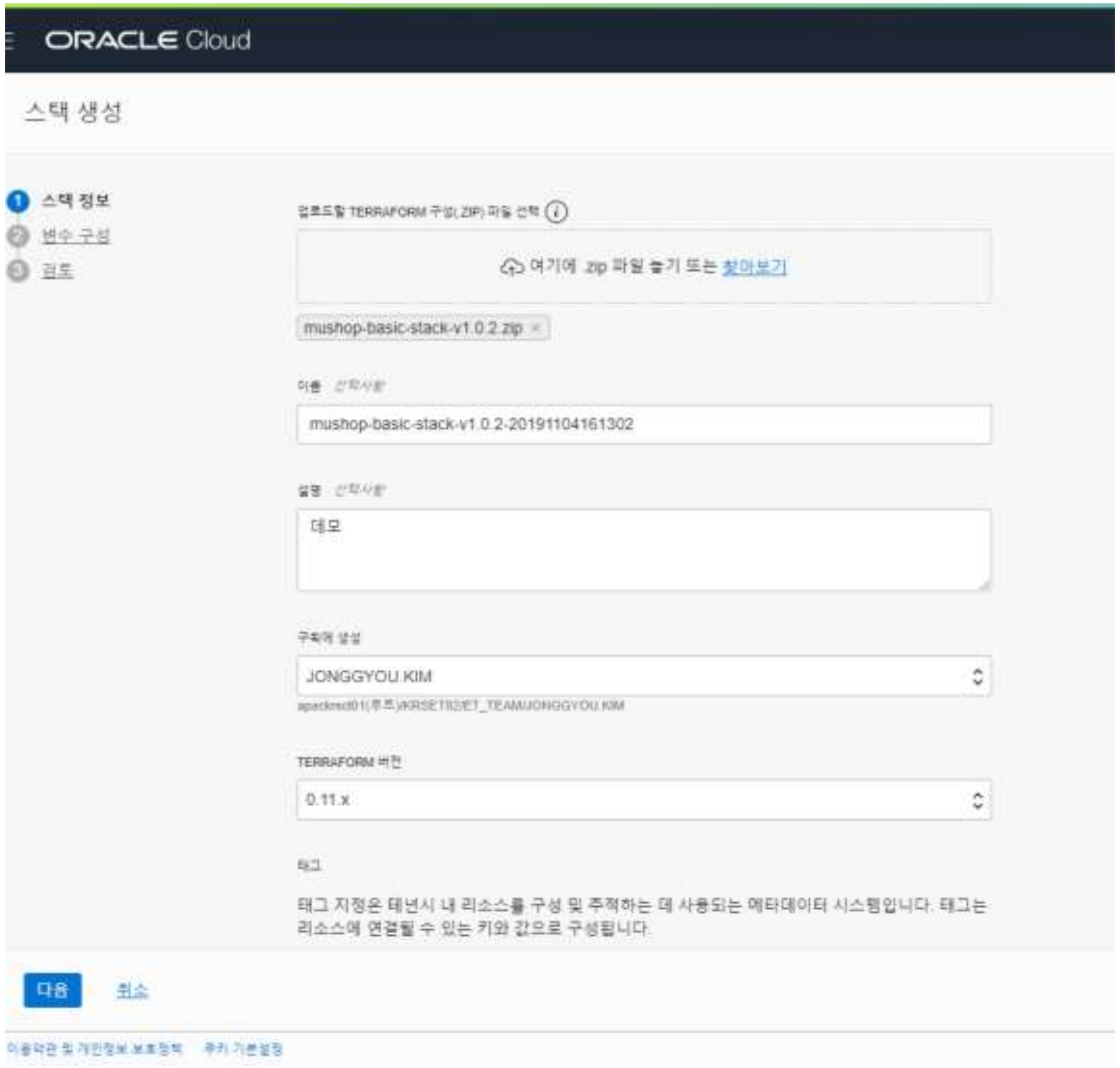


Resource Manager

Workflow : Step1



- 리소스에 대한 Terraform 구성 만들기
- 선택적 Terraform 모듈 작성
- Terraform 파일을 포함하는 zip 파일 만들기



스택 생성

1 스택 정보
2 변수 구성
3 검토

업로드할 TERRAFORM 구성(.ZIP) 파일 선택

여기에 .zip 파일 놓기 또는 찾아보기

mushop-basic-stack-v1.0.2.zip ×

이름 선택사항

mushop-basic-stack-v1.0.2-20191104161302

설명 선택사항

데모

구획에 생성

JONGGYOU.KIM

apackmcd01(루트)/KRSETB2/ET_TEAM/JONGGYOU.KIM

TERRAFORM 버전

0.11.x

태그

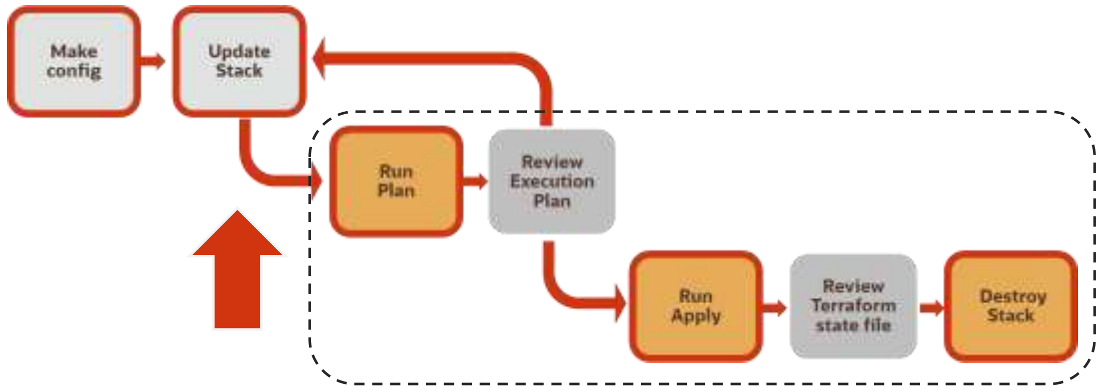
태그 지정은 테넌시 내 리소스를 구성 및 추적하는 데 사용되는 메타데이터 시스템입니다. 태그는 리소스에 연결될 수 있는 키와 값으로 구성됩니다.

다음 취소

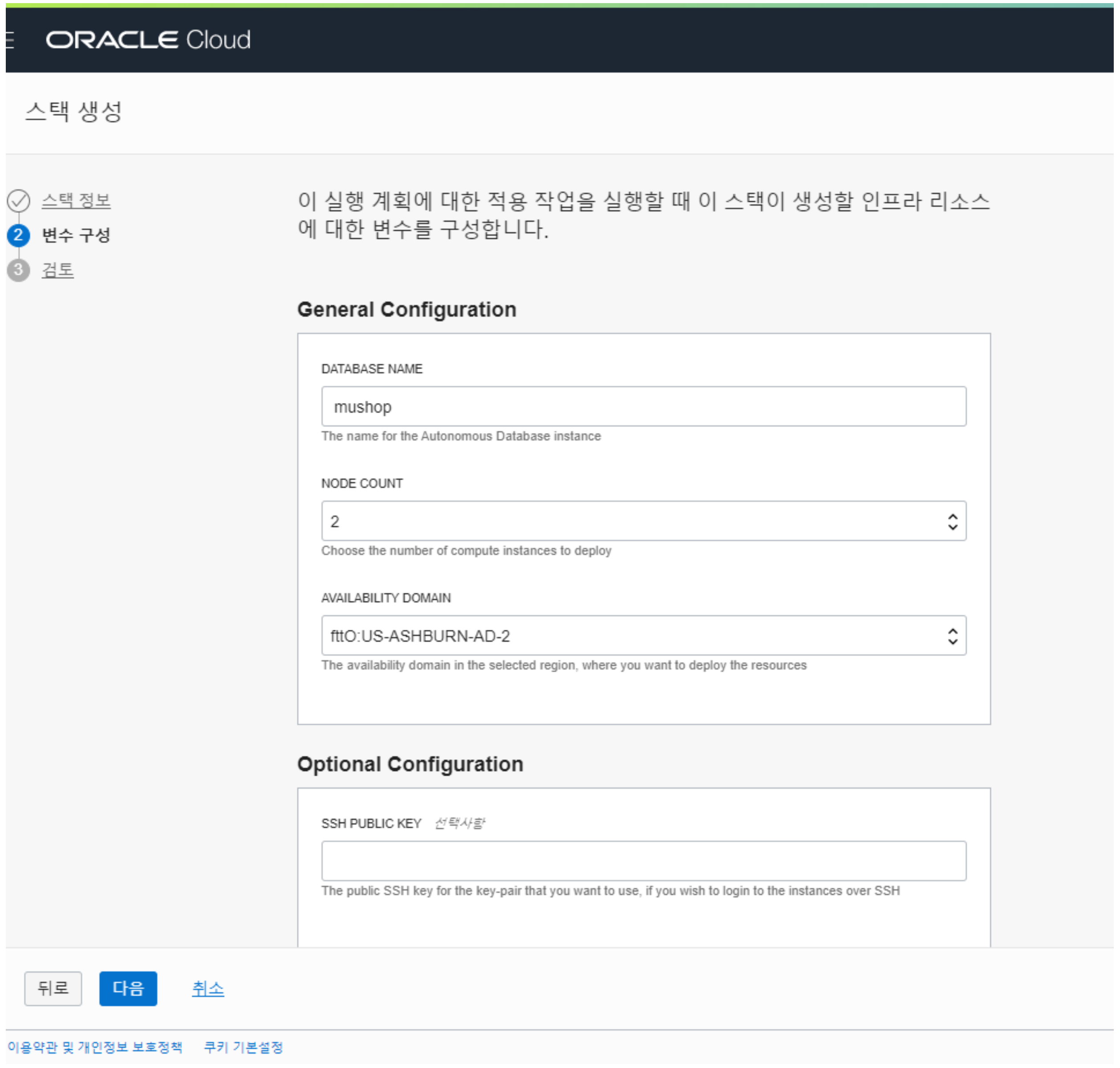
이용약관 및 개인정보 보호정책 쿠키 기본설정

Resource Manager

Workflow : Step2



- 구획 내에서 관리하는 일련의 리소스
- Terraform 구성 파일 및 Terraform 상태 파일에 매핑



스택 생성

이 실행 계획에 대한 적용 작업을 실행할 때 이 스택이 생성할 인프라 리소스에 대한 변수를 구성합니다.

General Configuration

DATABASE NAME
mushop
The name for the Autonomous Database instance

NODE COUNT
2
Choose the number of compute instances to deploy

AVAILABILITY DOMAIN
fttO:US-ASHBURN-AD-2
The availability domain in the selected region, where you want to deploy the resources

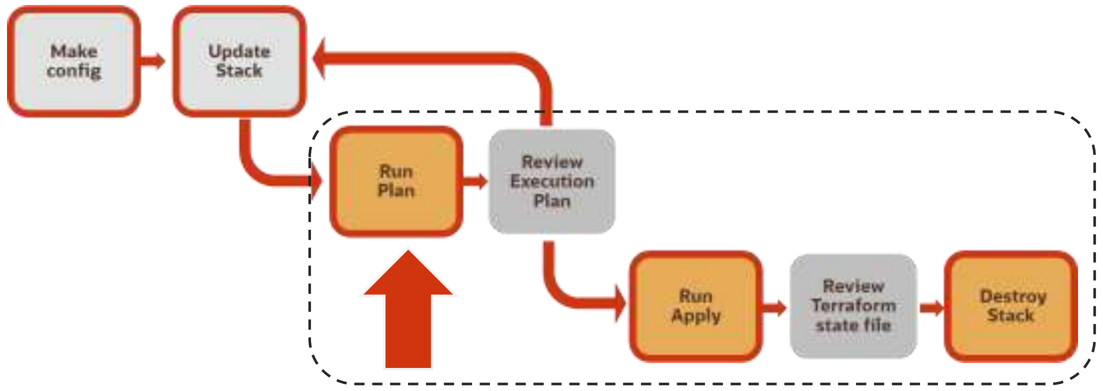
Optional Configuration

SSH PUBLIC KEY *선택사항*

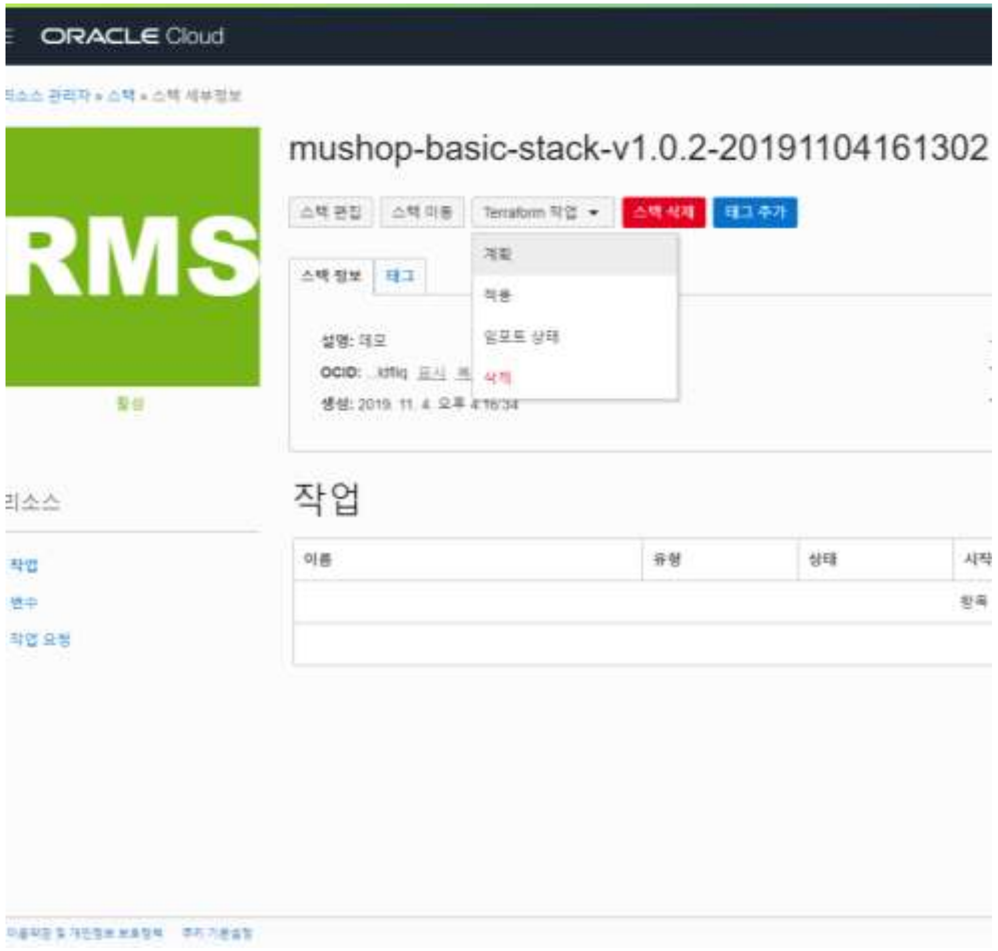
The public SSH key for the key-pair that you want to use, if you wish to login to the instances over SSH

Resource Manager

Workflow : Step3



- 스택에 대해 실행되는 Terraform
- 계획(Plan), 적용(Apply), 삭제(Delete)



mushop-basic-stack-v1.0.2-20191104161302

설명: 데모
OCID: ...ktfiq 표시 복사
생성: 2019. 11. 4. 오후 4:16:34

작업

LIVE DEMO

- Resource Manager

64

Reference

- https://www.slideshare.net/LiorKamrat/infrastructure-as-code-getting-started-concepts-tools
- https://www.slideshare.net/zekelabs/02-terraform-core-concepts?from_action=save
- https://learning.oreilly.com/library/view/terraform-up/9781492046899/
- https://www.44bits.io/ko/post/terraform_introduction_infrastrucute_as_code

Thank You