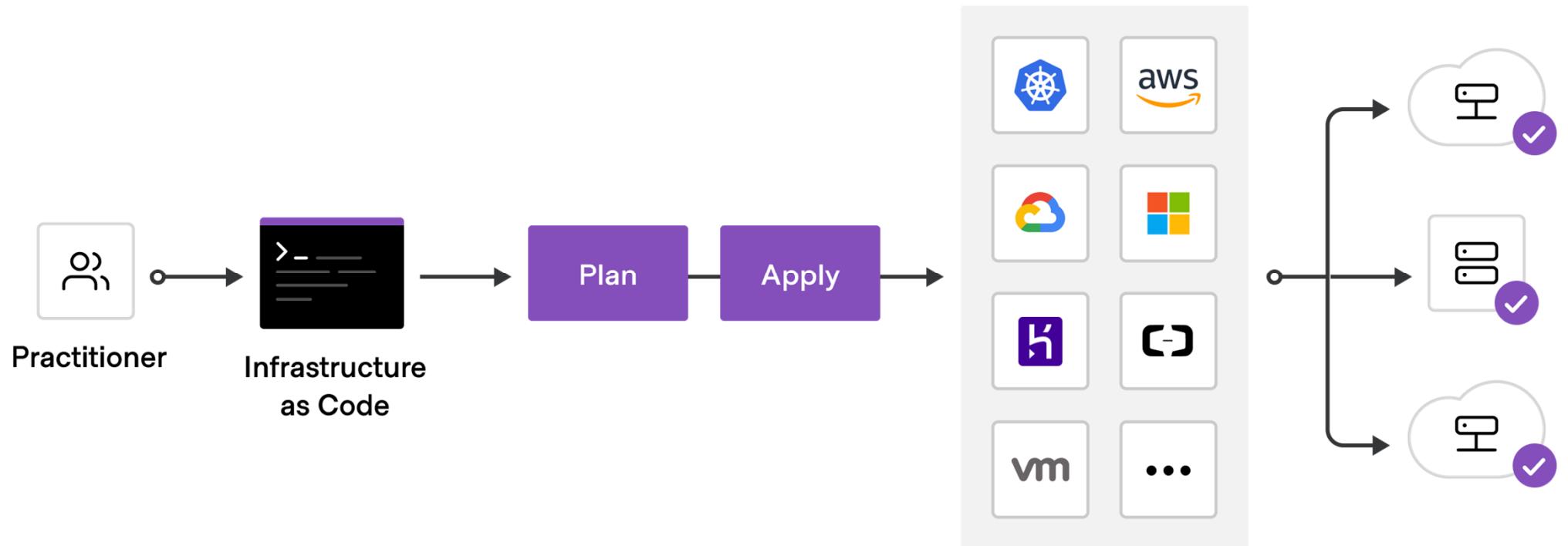


03

Infrastructure as Code

Terraform

Schéma général



Terraform

Comment l'utiliser ?

— Les commandes en mode CLI

- Interface ligne de commande s'utilise en tapant « `terraform option` » ou option est :
 - version => récupère la version du client Terraform et l'architecture système
 - init => initialisation et reconnaissance du provider et de l'espace d'application (projet/tenant) des autres commandes
 - plan => visualisation (ou écriture si output) du plan d'exécution permettant les vérifications avant apply.
 - apply => applique le contenu ou le différentiel (si un apply à déjà été fait) des fichiers TF.
 - destroy => supprime les éléments créé en regard du contenu du TfState
 - Graph => Graphique des dépendances.

Terraform

Initialisation de Terraform et déclaration Provider / Région

Initialisation de Terraform et des provider

- └ Indiquer le provider requis lors de l'init de l'environnement (source et version)
- └ Indiquer la version de Terraform

Déclarer le provider AWS

- └ Préciser la région de disponibilité des instances
- └ US East (N. Virginia) → us-east-1

```
terraform {  
    required_providers {  
        aws = {  
            source  = "hashicorp/aws"  
            version = "~> 4.16"  
        }  
    }  
  
    required_version = ">= 1.2.0"  
}  
  
provider "aws" {  
    region = "us-east-1"  
}
```

Terraform

Création d'un groupe de sécurité + ports entrée

— Création du groupe de sécurité

- └ aws_security_group : type de la ressource
- └ server-sg : nom unique de la ressource dans Terraform
- └ Name : nom du SG dans l'interface AWS
- └ Description : description du SG
- └ vpc_id : le projet auquel rattaché le SG
- └ Tag / Name : Nom de la ressource dans AWS

— Ouverture des ports d'entrée (80 sur TCP)

- └ Ouvrir une balise Ingress
- └ Spécifier port début -> port fin
- └ Préciser protocole : tcp, udp, icmp
- └ Indiquer le réseau source : cidr_blocks

```
resource "aws_security_group" « server-sg" {  
    name      = « srv-sg-e01"  
    description = "Allow inbound traffic to My Server"  
    vpc_id     = "vpc-0777f2a2c8769e96d"  
  
    ingress {  
        description      = « HTTP from VPC"  
        from_port        = 80  
        to_port          = 80  
        protocol         = "tcp"  
        cidr_blocks     = ["0.0.0.0/0"]  
    }  
  
    tags = {  
        Name = « MonServer-sg-e01"  
    }  
}
```

Terraform

Création d'un groupe de sécurité + ports sortie

— Création du groupe de sécurité

- └ aws_security_group : type de la ressource
- └ server-sg : nom unique de la ressource dans Terraform
- └ Name : nom du SG dans l'interface AWS
- └ Description : description du SG
- └ vpc_id : le projet auquel rattaché le SG
- └ Tag / Name : Nom de la ressource dans AWS

— Ouverture des ports de sortie (443 sur tous les protocoles)

- └ Ouvrir une balise Egress
- └ Spécifier port début -> port fin
0 pour tous les ports
- └ Préciser protocole : tcp, udp, icmp, -1
- └ Indiquer le réseau destination : cidr_blocks

```
resource "aws_security_group" "all-sg" {  
    name      = "allserver-sg-e01"  
    description = "Allow outbound traffic to all  
servers"  
    vpc_id     = "vpc-0777f2a2c8769e96d"  
  
    egress {  
        from_port      = 443  
        to_port        = 443  
        protocol       = "-1"  
        cidr_blocks    = ["0.0.0.0/0"]  
    }  
  
    tags = {  
        Name = "AllServer-out-sg-e01"  
    }  
}
```

Terraform

Création d'une instance

— Création de l'instance EC2

- └ aws_instance : type de la ressource
- └ Api_server : nom unique de la ressource dans Terraform
- └ ami : identifiant de l'image utilisée pour créer l'image
- └ Instance_type : gabarit de l'instance (RAM, vcpu)
- └ vpc_security_group_ids : liste des groupes de sécurité à appliquer
- └ User_data : script de personnalisation de l'instance au premier déploiement
- └ Tag / Name : Nom de la ressource dans AWS

```
resource "aws_instance" "api_server" {  
    ami          = "ami-0493936afbe820b28"  
    instance_type = "t2.micro"  
    vpc_security_group_ids = [aws_security_group.all-sg.id,aws_security_group.api-sg.id]  
    user_data = "${file("user-data-api.sh")}"  
  
    tags = {  
        Name = "API_Server_E01"  
    }  
}
```

Terraform

Récupération d'un output

— Chargement de l'id d'une instance

- └ output : indique qu'on interroge une donnée du provider
- └ Value : chemin d'accès à la ressource

— Chargement de l'adresse ip publique d'une instance

- └ output : indique qu'on interroge une donnée du provider
- └ Value : chemin d'accès à la ressource

```
output "instance_mon_server_id" {  
    description = "ID of the EC2 instance"  
    value      = aws_instance.api_server.id  
}  
  
output "instance_api_server_public_ip" {  
    description = "Public IP address of the EC2  
    instance API"  
    value      = aws_instance.api_server.public_ip  
}
```