



## EVALUATION EN COURS DE FORMATION

### PARCOURS ADMINISTRATEUR SYSTEME DEVOPS



## Mettre l'Infrastructure en Production dans le Cloud

Evaluation : ECF4

Version : du 10/11/2025

Auteur : FT

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

## Sommaire

### Table des matières

<b>1. PRESENTATION .....</b>	<b>2</b>
1.1. DESCRIPTION DE LA COMPETENCE – PROCESSUS DE MISE EN ŒUVRE .....	2
1.2. CONTEXTE(S) PROFESSIONNEL(S) DE MISE EN ŒUVRE.....	2
1.3. CRITERES DE PERFORMANCE.....	2
1.4. CRITERES DE PERFORMANCE.....	2
1.5. CAHIER DES CHARGES :.....	2
<b>2. ECF1 - QUESTIONS.....</b>	<b>3</b>
2.1. QU'EST-CE QU'UN VPC ? .....	3
2.2. POURQUOI UTILISER UN VPC ? .....	3
2.3. POURQUOI DEVONS-NOUS UTILISER UNE INTERNET GATEWAY LORS DE L'USAGE D'UN VPC.....	4
2.4. POURQUOI UTILISER UNE TABLE DE ROUTAGE ?.....	5
2.5. POURQUOI EST-IL UNE BONNE PRATIQUE D'UTILISER UN SECURITY GROUP ? .....	5
2.6. PEUT-ON ACCEDER AUX RESSOURCES D'UN VPC SANS INTERNET GATEWAY?.....	6
<b>3. INFRASTRUCTURE EN PRODUCTION.....</b>	<b>7</b>
3.1. ETAPES DE CREATION .....	7
3.2. CREATION DES MODULES (MAIN.TF EN ANNEXE 4.1).....	7
3.3. ETAPES DE DEPLOIEMENT.....	17
3.4. VERIFICATION EN PRODUCTION .....	19
3.5. COMMIT CHANGES .....	20
<b>4. ANNEXE .....</b>	<b>22</b>
4.1. MAINT.TF .....	22
4.2. TERRAFORM PLAN .....	27
4.3. TROUBLESHOOTING.....	31

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 1. PRESENTATION

---

### 1.1. DESCRIPTION DE LA COMPETENCE – PROCESSUS DE MISE EN ŒUVRE

A la demande de son responsable, configurer une offre cloud afin de pouvoir héberger l'infrastructure.

La déployer à l'aide d'un outil de gestion des configurations et vérifier que chaque serveur fonctionne et que tous les services sont accessibles par les utilisateurs finaux.

### 1.2. CONTEXTE(S) PROFESSIONNEL(S) DE MISE EN ŒUVRE

Dans certains cas, l'ASD peut choisir lui-même l'offre cloud la plus adaptée et la proposer à son responsable.

### 1.3. CRITERES DE PERFORMANCE

- Les serveurs déployés sont fonctionnels
- Les serveurs sont accessibles par les utilisateurs finaux
- L'architecture déployée est documentée

### 1.4. CRITERES DE PERFORMANCE

- Prendre en compte l'offre de services d'un fournisseur de services cloud
- Paramétrer les services cloud retenus
- Utiliser un outil de gestion des configurations de type Ansible ou Terraform
- Automatiser le déploiement de l'infrastructure avec l'outil de gestion des configurations
- Documenter le déploiement
- Diagnostiquer un dysfonctionnement et le corriger
- Dialoguer avec le fournisseur de services cloud
- Consulter de la documentation technique rédigée en anglais
- Effectuer une veille technologique
- Connaissance des offres cloud (IaaS, PaaS et SaaS)
- Connaissance des principes de la réversibilité
- Connaissance de l'architecture d'un réseau cloud
- Connaissance des recommandations de sécurité de l'ANSSI

### 1.5. CAHIER DES CHARGES :

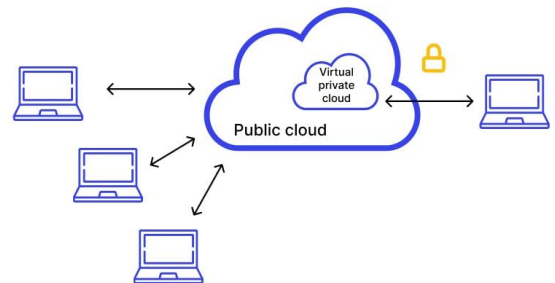
- **IMPORTANT** : Toujours Destroy après avoir Apply pour ne pas payer quoi que ce soit.
- Le cahier des charges est réalisé avec une terminologie AWS.

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 2. ECF1 - QUESTIONS

### 2.1. QU'EST-CE QU'UN VPC ?

- Un cloud privé virtuel (VPC) est une **section sécurisée et isolée d'un cloud public** qui permet aux entreprises d'exécuter leurs charges de travail avec un contrôle et une confidentialité accrus.
- Contrairement aux réseaux locaux traditionnels, un VPC offre un réseau cloud avec des paramètres personnalisables, permettant aux entreprises de définir leurs propres plages d'adresses IP, sous-réseaux, tables de routage et politiques de sécurité.
- Un VPC fonctionne comme un centre de données privé au sein de l'infrastructure d'un fournisseur cloud, tel qu'AWS, Google Cloud ou Microsoft Azure.
- Grâce à un VPC, les entreprises peuvent héberger des applications, des bases de données et d'autres ressources tout en maintenant des contrôles d'accès stricts.
- L'isolation au sein d'un environnement de cloud public garantit que les données et les charges de travail restent séparées de celles des autres clients, réduisant ainsi le risque d'accès non autorisé.



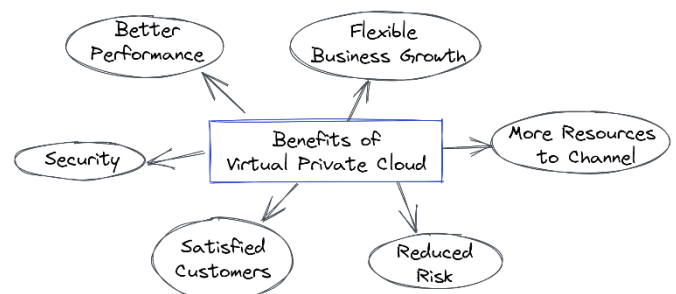
### 2.2. POURQUOI UTILISER UN VPC ?

- L'une des principales caractéristiques d'un VPC est **l'isolation du réseau**.

Contrairement aux environnements de cloud public traditionnels, où plusieurs clients partagent les ressources réseau, un VPC permet aux entreprises de créer un espace réseau dédié.

- Les VPC offrent des contrôles de sécurité robustes pour protéger les données et les applications.

Les **groupes de sécurité et les listes de contrôle d'accès réseau (ACL)** permettent aux administrateurs de définir des règles qui contrôlent le trafic entrant et sortant à différents niveaux.



- Un Cloud Privé Virtuel offre un ensemble complet de fonctionnalités qui en font une **solution sécurisée, flexible et évolutive** pour les entreprises souhaitant exploiter le Cloud Computing.

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 2.3. POURQUOI DEVONS-NOUS UTILISER UNE INTERNET GATEWAY LORS DE L'USAGE D'UN VPC.

Voici déjà les points à établir pour accéder à Internet directement depuis une instance à l'intérieur d'un VPC :

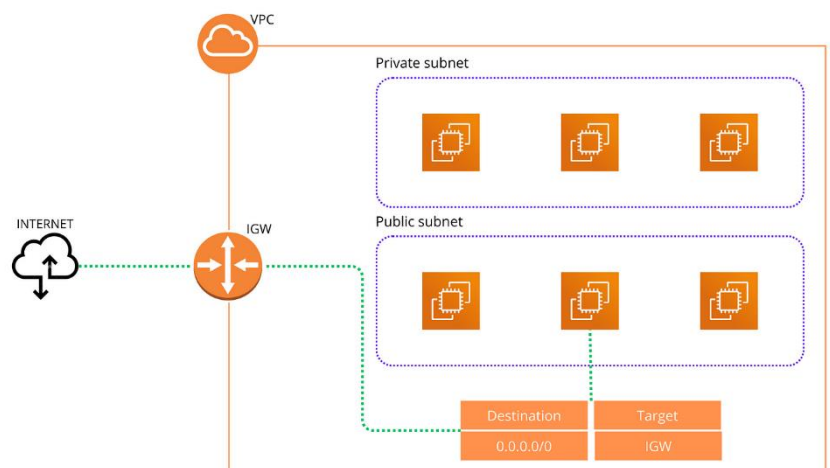
- Une **passerelle Internet (IGW)** connectée au VPC
- Un **sous-réseau avec une table de routage** qui a une route par défaut (0.0.0.0/0) via l'IGW (appelé « sous-réseau public »)
- Une **IP publique ou une IP élastique** attachée à l'instance (adresse IP publique fixe qu'on peut associer à des ressources cloud)
- Un **groupe de sécurité** attaché à l'instance qui autorise le trafic sortant
- **Listes de contrôle d'accès réseau** associées au sous-réseau qui autorise le trafic

Une passerelle IGW effectue une sorte de NAT 1 à 1 vers les adresses IP publiques mappées aux instances (le NAT permet aux instances dans un subnet privé **d'accéder à internet sans être exposées directement**).

- La passerelle Internet permet aux ressources (comme les instances EC2) des sous-réseaux publics de se connecter à Internet.

De même, les ressources sur Internet peuvent établir une connexion aux ressources du sous-réseau via le réseau public.

- Si un VPC ne dispose pas de passerelle Internet, les ressources du VPC ne sont pas accessibles depuis Internet (à moins que le trafic ne circule via un réseau d'entreprise et un VPN/une connexion directe).



Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

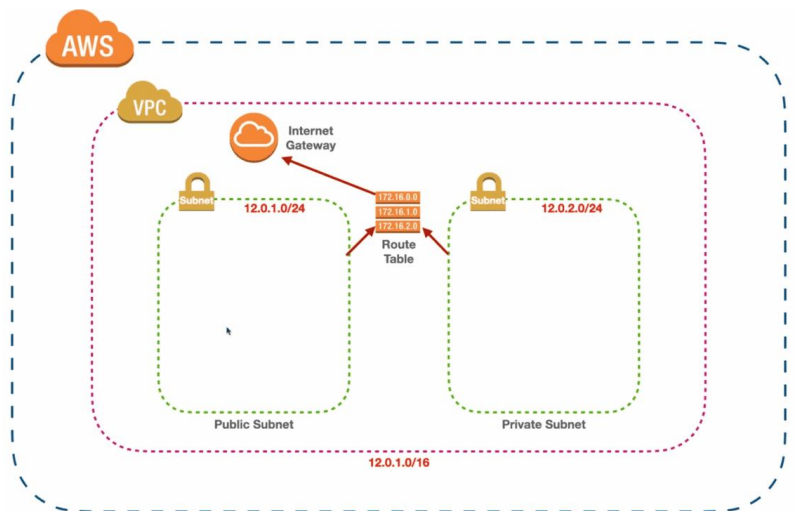
## 2.4. POURQUOI UTILISER UNE TABLE DE ROUTAGE ?

Le rôle d'une table de routage est de **diriger le trafic réseau en fonction de l'adresse IP de destination**.

- Chaque route de la table spécifie une destination (sous la forme d'une adresse IP ou d'un bloc CIDR) et une cible (comme une passerelle Internet (IGW), une interface réseau ou une autre table de routage).
- Une route locale est un type de route spécifique qui permet la communication au sein du VPC.

Par exemple, si on dispose d'un sous-réseau avec une table de routage associée dont une route pointe vers une passerelle Internet (IGW), le trafic de ce sous-réseau vers Internet est autorisé.

De même, le trafic provenant d'un sous-réseau avec une route pointant vers une passerelle de transit est dirigé vers le réseau approprié.



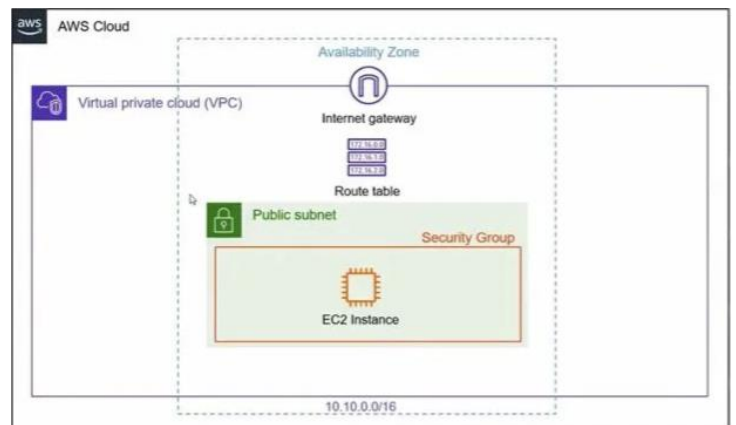
## 2.5. POURQUOI EST-IL UNE BONNE PRATIQUE D'UTILISER UN SECURITY GROUP ?

Les groupes de sécurité sont des éléments fondamentaux de la sécurité dans un environnement Cloud.

Ils sont similaires aux pare-feu, mais ne sont pas identiques, et sont un ensemble de règles qui contrôlent le trafic entrant et sortant des instances, car il est nécessaire d'avoir un filtrage de couche 3 (Réseau) et 4 (Transport) dans le cloud, comme dans le monde physique.

**Caractéristiques principales :**

- Si un trafic entrant est autorisé, la réponse sortante est automatiquement autorisée, et vice versa.
- Un groupe de sécurité est assigné aux interfaces réseau (ex: à une instance EC2).
- Une ressource peut avoir plusieurs groupes de sécurité attachés.
- Changer une règle dans un groupe de sécurité affecte instantanément toutes les ressources qui utilisent ce groupe.



Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

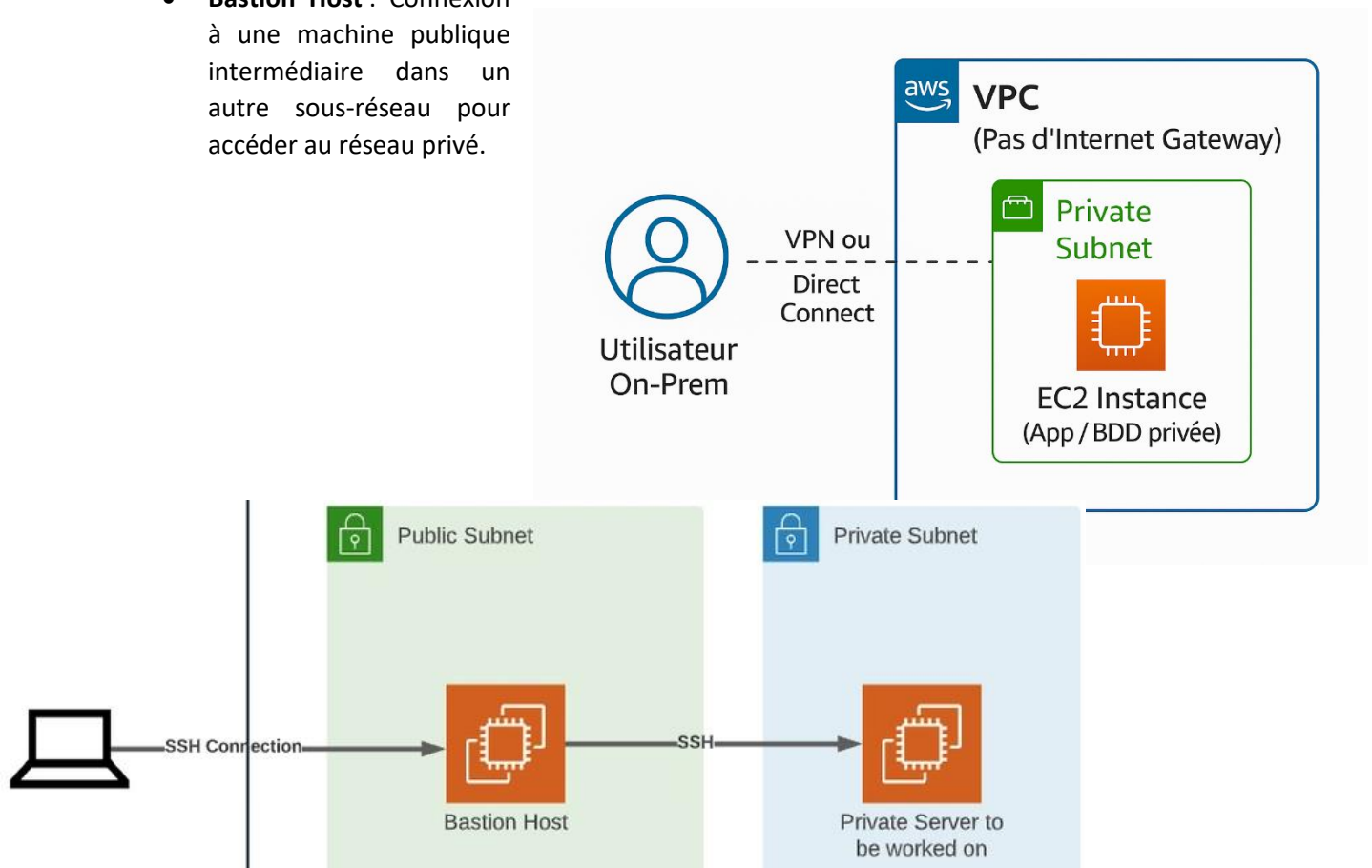
## 2.6. PEUT-ON ACCEDER AUX RESSOURCES D'UN VPC SANS INTERNET GATEWAY?

On peut accéder aux ressources d'un VPC sans Internet Gateway, mais pas directement depuis Internet.

Une Internet Gateway (IGW) sert à rendre accessible des ressources comme EC2 depuis Internet (IP publique, trafic HTTP, SSH, etc.).

Il existe des méthodes pour accéder aux instances sans IGW comme :

- **VPN** : Connexion du réseau privé local au VPC via un tunnel sécurisé sans passer par Internet.
- **Direct Connect** : Connexion dédiée entre entreprise et fournisseur cloud
- **Bastion Host** : Connexion à une machine publique intermédiaire dans un autre sous-réseau pour accéder au réseau privé.





Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 3. INFRASTRUCTURE EN PRODUCTION

### 3.1. ETAPES DE CREATION

1. [Un VPC avec un bloc CIDR spécifique](#)
2. [Une internet gateway IGW relié à ce VPC](#)
3. [Une route table permettant de router tout le trafic provenant de la gateway](#)
4. [Un SUBNET avec un bloc CIDR spécifique](#)
5. [Une association route table / subnet](#)
6. [Un SECURITY GROUP permettant la connection SSH, HTTP et HTTPS](#)
7. [Une INTERFACE NETWORK et une ELASTIC IP](#)
8. [Une INSTANCE DEBIAN bullseye liée au network interface](#)
9. [Une KEY PAIR reliée à l'instance permettant de se connecter en ssh](#)
10. [Et enfin un script exécuté dans les user\\_data de l'instance configurant un server APACHE et injectant notre nom et prénom dans le fichier /var/www/html/index.html](#)
11. [OUTPUT.TF](#) (des Modules)
12. [VARIABLES.TF](#) (des Modules)

### 3.2. CREATION DES MODULES (MAIN.TF EN [ANNEXE 4.1](#))

terraformcloud-aws-vm/

```

├─ main.tf
├─ variables.tf
├─ terraform.tfvars
├─ outputs.tf
├─ index.html
├─ .vault-token
├─ modules/
| └─ ec2/           # Contient création Instance Debian
|   ├── main.tf
|   ├── output.tf
|   └── variables.tf
|
| └─ eip/           # Contient création Interface réseau et Adresse IP Elastic
|   ├── main.tf
|   ├── output.tf
|   └── variables.tf
|
| └─ sg/            # Contient création Security Group
|   ├── main.tf
|   ├── output.tf
|   └── variables.tf
|
| └─ vpc/           # Contient création VPC/Gateway/Route/Subnet
|   ├── main.tf
|   └── output.tf

```



Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

## 1. Création VPC avec un bloc CIDR spécifique

```
resource "aws_vpc" "vpc_1" {
  cidr_block = "192.168.0.0/16"      # Bloc CIDR que vous voulez utiliser pour votre VPC
  enable_dns_support = true          # Active la prise en charge DNS dans le VPC
  enable_dns_hostnames = true        # Permet la création de noms d'hôtes DNS dans le VPC
  tags = {
    Name = "aws_vpc_1"               # Identification de la ressource dans la console AWS
  }
}
```

## 2. Création Internet Gateway IGW reliée à ce VPC

```
resource "aws_internet_gateway" "igw_1" {
  vpc_id = aws_vpc.vpc_1.id          # Associe l'Internet Gateway à vpc_1
  tags = {
    Name = "InternetGateway_1"       # Identification de la ressource dans la console AWS
  }
}
```

## 3. Création Table de routage

```
resource "aws_route_table" "route_table_1" {
  vpc_id = aws_vpc.vpc_1.id          # Associe cette table de routage à vpc_1
  route {
    cidr_block = "0.0.0.0/0"         # Cette route permet l'accès à Internet
    gateway_id = aws_internet_gateway.igw_1.id # Trafic destiné à l'extérieur du VPC envoyé à IGW
  }
  tags = {
    Name = "RouteTable_1"            # Identification de la ressource dans la console AWS
  }
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

#### 4. Création SUBNET avec un bloc CIDR spécifique

```
resource "aws_subnet" "subnet_1" {
  vpc_id = aws_vpc.vpc_1.id           # Associe ce sous-réseau à vpc_1
  cidr_block = "192.168.1.0/24"       # Bloc CIDR du sous-réseau
  availability_zone = "eu-east-1"     # Zone de disponibilité cible
  tags = {
    Name = "Subnet_1"                 # Identification de la ressource dans la console AWS
  }
}
```

#### 5. Création Association route table/subnet

```
resource "aws_route_table_association" "route_table_association_1" {
  subnet_id = aws_subnet.subnet_1.id      # ID du sous-réseau 192.168.1.0/24
  route_table_id = aws_route_table.route_table_1.id # ID de la table de routage
}
```

#### 6. Création SECURITY GROUP (connection SSH, HTTP et HTTPS)

```
resource "aws_security_group" "security_group_1" {
  name = "security-group-1"
  description = "Security group to allow SSH, HTTP, and HTTPS"
  vpc_id = var.vpc_id           # Associe ce security group à vpc_1
}
```

# Autoriser les connexions SSH (port 22)

```
ingress {
  from_port = 22
  to_port = 22
  protocol = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

# Autoriser les connexions HTTP (port 80)

```
ingress {
  from_port = 80
  to_port   = 80
  protocol  = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}
```

# Autoriser les connexions HTTPS (port 443)

```
ingress {
  from_port = 443
  to_port   = 443
  protocol  = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}
```

# Autoriser le trafic sortant

```
egress {
  from_port = 0
  to_port   = 0
  protocol  = "-1"
  cidr_blocks = ["0.0.0.0/0"]
}
```

# Autorise tout le trafic sortant

```
tags = {
  Name = "SecurityGroup_1"
}
```

# Identification de la ressource dans la console AWS

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

## 7. Création INTERFACE NETWORK et ELASTIC IP

```
resource "aws_eip" "eip_1" {
  domain = "vpc" # Permet l'attache à une ENI dans un VPC
  tags = {
    Name = "ElasticIP_1" # Identification de la ressource dans la console AWS
  }
}
```

### # Création Network Interface


```
resource "aws_network_interface" "network_interface_1" {
  subnet_id = var.subnet_id # Associe cette interface au sous-réseau subnet_1
  private_ips = ["192.168.1.20"] # Adresse IP privée spécifique
  security_groups = [var.security_group_id]
  tags = {
    Name = "NetworkInterface_1" # Identification de la ressource dans la console AWS
  }
}
```

### # Attacher l'Elastic IP (EIP) à Network Interface (ENI)

```
resource "aws_eip_association" "eip_assoc_1" {
  allocation_id = aws_eip.eip_1.id # ID de l'IP élastique (EIP) associée à l'interface réseau
  network_interface_id = aws_network_interface.network_interface_1.id # ID de l'interface réseau (ENI) associer l'EIP
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 8. Création Instance DEBIAN Bullseye liée au network interface



**Debian 11**  
Debian [\[link\]](#)  
★★★★★ 3 évaluations AWS [\[link\]](#) | 168 évaluations externes [\[link\]](#)  
Free Tier

Présentation
Informations sur le produit
Tarification
Utilisation

Une partie de cette page a été traduite automatiquement.

Debian: The universal operating system

Prix total standard

0,01 \$US/Hr

Latest version

debian-11-amd64-20250402-2070

Présentation

Informations sur le produit

Tarification

Utilisation

Support

### 64-bit (x86) Amazon Machine Image (AMI)

Instructions d'utilisation

After launching your instance, connect to it using a Secure Shell (SSH) client with the SSH key you specified at launch. The default username is 'admin'.

<https://wiki.debian.org/Cloud/AmazonEC2Image/Bullseye> :

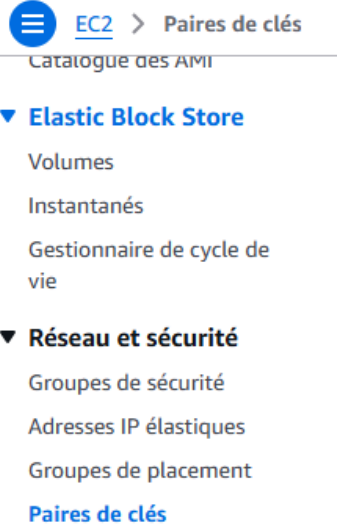
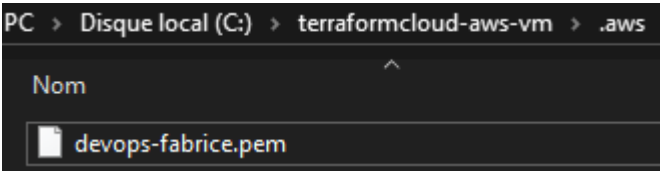
Region	AMD64 AMI ID
us-east-1	ami-000a08b963606bb82

```
resource "aws_instance" "debian_bullseye_1" {
  ami      = "ami-000a08b963606bb82" # debian-11-amd64-20250402-2070 sur us-east-1
  instance_type = "t2.micro"           # taille de l'instance EC2 (1 vCPU/ 1 GB)
  key_name   = "devops-fabrice"       # Nom de la paire de clés SSH créée dans AWS.
  network_interface {
    device_index = 0                  # Spécifie l'index de l'interface réseau
    # L'instance Debian utilisera l'interface réseau network_interface_1
    network_interface_id = var.network_interface_id
  }
  tags = {
    Name = "DebianBullseye_1"        # Identification de la ressource dans la console AWS
  }
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 9. Création KEY PAIR reliée à l'instance (connexion SSH)

<https://console.aws.amazon.com/ec2>

<ul style="list-style-type: none"> <li>Réseau et Sécurité/Paire de Clés</li> </ul>							
<ul style="list-style-type: none"> <li>Créer Paire de clés <ul style="list-style-type: none"> <li>Type : RSA</li> <li>Format : PEM</li> </ul> </li> <li>Téléchargement fichier clé privé</li> </ul> <p><b>devops-fabrice.pem</b> qui sera stocké dans dossier caché <b>.aws</b> dans <b>C:\terraformcloud-aws-vm</b></p> 	<h3>Créer une paire de clés</h3> <div data-bbox="1018 1128 1364 1568"> <p><b>Paire de clés</b> Une paire de clés, composée d'une clé publique et d'une clé privée.</p> <p><b>Nom</b> <input type="text" value="devops-fabrice"/> <small>Le nom peut avoir un maximum de 255 caractères.</small></p> <p><b>Type de paire de clés</b> <a href="#">Informations</a> <input checked="" type="radio"/> RSA</p> <p><b>Format de fichier de clé privée</b> <input checked="" type="radio"/> .pem <small>À utiliser avec OpenSSH</small></p> </div>						
<div data-bbox="443 1608 1233 1832"> <p><b>Paires de clés (1)</b> <a href="#">Informations</a></p> <p><input type="text" value="Rechercher Paire de clés par attribut ou identification"/></p> <table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>Nom</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>devops-fabrice</td> <td>rsa</td> </tr> </tbody> </table> </div>		<input type="checkbox"/>	Nom	Type	<input type="checkbox"/>	devops-fabrice	rsa
<input type="checkbox"/>	Nom	Type					
<input type="checkbox"/>	devops-fabrice	rsa					

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

## 10. Configuration Serveur APACHE

- A rajouter dans le bloc "**aws\_instance**" et sera exécuté automatiquement à la première création de l'instance EC2.

Le script "**apache\_install.sh**" sera stocké dans le répertoire du module EC2.

```
resource "aws_instance" "debian_bullseye_1" {
  user_data = file("${path.module}/apache_install.sh")
}
```

- Contenu fichier **apache\_install.sh** :

```
#!/bin/bash

apt update
apt install -y apache2
systemctl enable apache2
systemctl start apache2

echo '<html><body><h1><span style="color: blue;">Nom</span>: DevOps <span style="color: blue;">Prenom</span>: Fabrice</h1></body></html>' > /var/www/html/index.html
```



Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

## 11. OUTPUT.TF (des Modules)

Valeurs après application de la configuration pouvant être utilisées par d'autres scripts ou modules.

### a) Module EC2

```
output "aws_instance_id" {
  value = aws_instance.debian_bullseye_1.id
}
```

### b) Module EIP

```
output "aws_eip_id" {
  value = aws_eip.eip_1.id
}

output "public_ip" {
  description = "Public IP of the Elastic IP"
  value       = aws_eip.eip_1.public_ip
}

output "aws_eni_id" {
  value = aws_network_interface.network_interface_1.id
}
```

- Enregistrement de l'IP dans un fichier texte : **maint.tf** (racine)

```
resource "null_resource" "export_public_ip" {
  depends_on = [module.eip]
  provisioner "local-exec" {
    command = "echo ${module.eip.public_ip} > public_ip.txt"
  }
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

### c) Module SG

```
output "aws_security_group_id" {
  value = aws_security_group.security_group_1.id
}
```

### d) Module VPC

```
output "aws_vpc_id" {
  value = aws_vpc.vpc_1.id
}

output "aws_subnet_id" {
  value = aws_subnet.subnet_1.id
}
```

## 12. VARIABLES.TF (des Modules)

### a) Module EC2

```
variable "security_group_id" {
  type = string
}

variable "network_interface_id" {
  type = string
}
```

### b) Module EIP

```
variable "subnet_id" {
  type = string
}

variable "security_group_id" {
  type = string
}
```

### c) Module SG

```
variable "vpc_id" {
  type = string
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>  Mettre l'Infrastructure en Production dans le Cloud	Auteur	FT
ECF4	Titre		Du	10/11/2025

### 3.3. ETAPES DE DEPLOIEMENT

#### a. Terraform INIT

```
$ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/null from the dependency lock file
- Reusing previous version of hashicorp/vault from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/null v3.2.4
- Using previously-installed hashicorp/vault v4.7.0
- Using previously-installed hashicorp/aws v5.95.0

Terraform has been successfully initialized!
```

#### b. Terraform VALIDATE

```
$ terraform validate
Success! The configuration is valid.
```

#### c. Terraform PLAN (en [ANNEXE 4.2](#))

```
$ terraform plan
data.vault_aws_access_credentials.creds: Reading...
data.vault_aws_access_credentials.creds: Still reading... [10s elapsed]
data.vault_aws_access_credentials.creds: Still reading... [20s elapsed]
data.vault_aws_access_credentials.creds: Read complete after 21s [id=aws/c

Terraform used the selected providers to generate the following execution
+ create
```

```
Plan: 11 to add, 0 to change, 0 to destroy.
```

Evaluation	Parcours	ADMINISTRATEUR SYSTEME DEVOPS Mettre l'Infrastructure en Production dans le Cloud	Auteur	FT
ECF4	Titre		Du	10/11/2025

#### d. Terraform APPLY

```

module.eip.aws_eip.eip_1: Creating...
module.vpc.aws_vpc.vpc_1: Creating...
module.eip.aws_eip.eip_1: Creation complete after 2s [id=eipalloc-07c53cb941e008d99]
module.vpc.aws_vpc.vpc_1: Still creating... [10s elapsed]
module.vpc.aws_vpc.vpc_1: Creation complete after 14s [id=vpc-0cd940139cfe3b3c4]
module.vpc.aws_internet_gateway.igw_1: Creating...
module.vpc.aws_subnet.subnet_1: Creating...
module.sg.aws_security_group.security_group_1: Creating...
module.vpc.aws_internet_gateway.igw_1: Creation complete after 1s [id=igw-0789cca12acb2b76d]
module.vpc.aws_route_table.route_table_1: Creating...
module.vpc.aws_subnet.subnet_1: Creation complete after 2s [id=subnet-046e84ed8f202ceae]
module.vpc.aws_route_table.route_table_1: Creation complete after 3s [id=rtb-0a79945d6bb0800aa]
module.vpc.aws_route_table_association.route_table_association_1: Creating...
module.vpc.aws_route_table_association.route_table_association_1: Creation complete after 1s [id=rtbassoc-03b67d6907869e263]
module.sg.aws_security_group.security_group_1: Creation complete after 5s [id=sg-078806b9a9cbfaa1c]
module.eip.aws_network_interface.network_interface_1: Creating...
module.eip.aws_network_interface.network_interface_1: Creation complete after 1s [id=eni-0b94a6adf2091f647]
module.eip.aws_eip_association.eip_assoc_1: Creating...
module.ec2.aws_instance.debian_bullseye_1: Creating...
module.eip.aws_eip_association.eip_assoc_1: Creation complete after 2s [id=eipassoc-0665fa73313bfb13b]
null_resource.export_public_ip: Creating...
null_resource.export_public_ip: Provisioning with 'local-exec'...
null_resource.export_public_ip (local-exec): Executing: ["cmd" "/C" "echo 54.147.118.63 > public_ip.txt"]
null_resource.export_public_ip: Creation complete after 0s [id=8602652922435011738]
module.ec2.aws_instance.debian_bullseye_1: Still creating... [10s elapsed]
module.ec2.aws_instance.debian_bullseye_1: Creation complete after 15s [id=i-0b468c4029436b2e6]

Apply complete! Resources: 11 added, 0 changed, 0 destroyed.

```

#### e. Terraform DESTROY

```

Plan: 0 to add, 0 to change, 11 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

null_resource.export_public_ip: Destroying... [id=8602652922435011738]
null_resource.export_public_ip: Destruction complete after 0s
module.eip.aws_eip_association.eip_assoc_1: Destroying... [id=eipassoc-0665fa73313bfb13b]
module.vpc.aws_route_table_association.route_table_association_1: Destroying... [id=rtbassoc-03b67d6907869e263]
module.ec2.aws_instance.debian_bullseye_1: Destroying... [id=i-0b468c4029436b2e6]
module.vpc.aws_route_table_association.route_table_association_1: Destruction complete after 2s
module.vpc.aws_route_table.route_table_1: Destroying... [id=rtb-0a79945d6bb0800aa]
module.eip.aws_eip_association.eip_assoc_1: Destruction complete after 3s
module.eip.aws_eip.eip_1: Destroying... [id=eipalloc-07c53cb941e008d99]
module.vpc.aws_route_table.route_table_1: Destruction complete after 1s
module.vpc.aws_internet_gateway.igw_1: Destroying... [id=igw-0789cca12acb2b76d]
module.vpc.aws_internet_gateway.igw_1: Destruction complete after 2s
module.eip.aws_eip.eip_1: Destruction complete after 2s
module.ec2.aws_instance.debian_bullseye_1: Still destroying... [id=i-0b468c4029436b2e6, 10s elapsed]
module.ec2.aws_instance.debian_bullseye_1: Still destroying... [id=i-0b468c4029436b2e6, 20s elapsed]
module.ec2.aws_instance.debian_bullseye_1: Still destroying... [id=i-0b468c4029436b2e6, 30s elapsed]
module.ec2.aws_instance.debian_bullseye_1: Still destroying... [id=i-0b468c4029436b2e6, 40s elapsed]
module.ec2.aws_instance.debian_bullseye_1: Still destroying... [id=i-0b468c4029436b2e6, 50s elapsed]
module.ec2.aws_instance.debian_bullseye_1: Still destroying... [id=i-0b468c4029436b2e6, 1m0s elapsed]
module.ec2.aws_instance.debian_bullseye_1: Destruction complete after 1m3s
module.eip.aws_network_interface.network_interface_1: Destroying... [id=eni-0b94a6adf2091f647]
module.eip.aws_network_interface.network_interface_1: Destruction complete after 1s
module.vpc.aws_subnet.subnet_1: Destroying... [id=subnet-046e84ed8f202ceae]
module.sg.aws_security_group.security_group_1: Destroying... [id=sg-078806b9a9cbfaa1c]
module.vpc.aws_subnet.subnet_1: Destruction complete after 1s
module.sg.aws_security_group.security_group_1: Destruction complete after 1s
module.vpc.aws_vpc.vpc_1: Destroying... [id=vpc-0cd940139cfe3b3c4]
module.vpc.aws_vpc.vpc_1: Destruction complete after 1s

Destroy complete! Resources: 11 destroyed.

```

Evaluation	Parcours	ADMINISTRATEUR SYSTEME DEVOPS Mettre l'Infrastructure en Production dans le Cloud	Auteur	FT
ECF4	Titre		Du	10/11/2025

### 3.4. VERIFICATION EN PRODUCTION

#### a. Instance Debian - EC2

Instances (1) Informations Date de la dernière mise à jour Il y a 1 minute Se connecter État de l'instance ▼ Actions ▼ Lancer des instanc

Rechercher Instance par attribut ou identification (case-sensitive) Tous les é... ▼

État de l'instance = running Effacer les filtres < 1

<input type="checkbox"/>	Name	ID d'instance	État de l'instance	Type d'i...	Zone de ...	DNS IPv4 public	Adresse IPv4...	IP élastique
<input type="checkbox"/>	DebianBullsey...	i-0b468c4029436b...	En cours d'ex...	t2.micro	us-east-1a	ec2-54-147-118-63.co...	54.147.118.63	54.147.118.63

#### b. Interface Réseau - ENI

Interfaces réseau (1) Informations less tha

Recherche

<input type="checkbox"/>	Name	ID d'interface réseau	ID de VPC	Zone de...	Noms des group.
<input type="checkbox"/>	NetworkInterf...	eni-0b94a6adf2091f647	vpc-0cd940139cfe3b3c4	us-east-1a	security-group-1

#### c. Adresse IP Elastic - EIP

##### Adresses IP Elastic (1)

Find elastic IP addresses by attribute or tag

<input type="checkbox"/>	Name	Adresse IPv4 allouée
<input type="checkbox"/>	ElasticIP_1	54.147.118.63

#### d. Security Group - SG

Groupes de sécurité (3) Informations Actions ▼ Exporter des groupes de sécurité au for

Rechercher des groupes de sécurité par attribut ou par balise

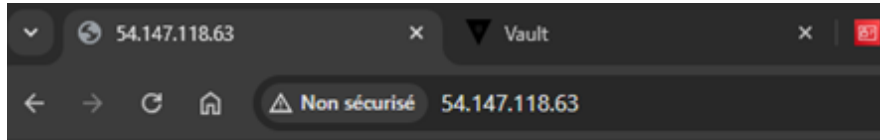
<input type="checkbox"/>	Name	ID du groupe de sécurité	Nom du groupe de sécurité	Description
<input type="checkbox"/>	SecurityGroup_1	sg-078806b9a9cbfaa1c	security-group-1	Security group to allow SSH, HTTP, and HTTPS

#### e. Réseau Privé Virtuel - VPC

<input type="checkbox"/>	Name	ID de VPC	État	CIDR IPv4
<input type="checkbox"/>	aws_vpc_1	vpc-0cd940139cfe3b3c4	Available	192.168.0.0/16

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>  Mettre l'Infrastructure en Production dans le Cloud	Auteur	FT
ECF4	Titre		Du	10/11/2025

## f. Serveur Apache – Page d'Accueil

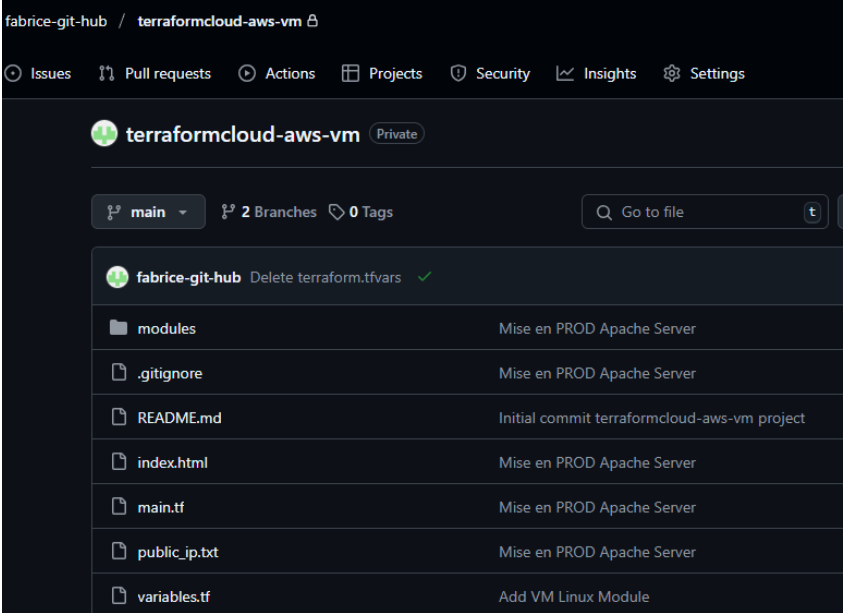


**Nom: DevOPs Prénom: Fabrice**

### 3.5. COMMIT CHANGES

git status	<pre>\$ git status On branch main Your branch is up to date with 'origin/main'.  Changes not staged for commit:   (use "git add/rm &lt;file&gt;..." to update what will be committed)   (use "git restore &lt;file&gt;..." to discard changes in working directory)         modified:   .gitignore         modified:   main.tf         deleted:    modules/vm_linux/main.tf  Untracked files:   (use "git add &lt;file&gt;..." to include in what will be committed)         index.html         modules/ec2/         modules/eip/         modules/sg/         modules/vpc/         public_ip.txt  no changes added to commit (use "git add" and/or "git commit -a")</pre>
git add .	<pre>\$ git status On branch main Your branch is up to date with 'origin/main'.  Changes to be committed:   (use "git restore --staged &lt;file&gt;..." to unstage)         modified:   .gitignore         new file:   index.html         modified:   main.tf         new file:   modules/ec2/apache_install.sh         new file:   modules/ec2/main.tf         new file:   modules/ec2/outputs.tf         new file:   modules/ec2/variables.tf         new file:   modules/eip/main.tf         new file:   modules/eip/outputs.tf         new file:   modules/eip/variables.tf         new file:   modules/sg/main.tf         new file:   modules/sg/outputs.tf         new file:   modules/sg/variables.tf         deleted:    modules/vm_linux/main.tf         new file:   modules/vpc/main.tf         new file:   modules/vpc/outputs.tf         new file:   public_ip.txt</pre>

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>  <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

<p><b>git commit -m "Mise en PROD Apache Server"</b></p>	<pre>\$ git commit -m "Mise en PROD Apache Server" [main 6b72c1b] Mise en PROD Apache Server 17 files changed, 220 insertions(+), 10 deletions(-) create mode 100644 index.html create mode 100644 modules/ec2/apache_install.sh create mode 100644 modules/ec2/main.tf create mode 100644 modules/ec2/outputs.tf create mode 100644 modules/ec2/variables.tf create mode 100644 modules/eip/main.tf create mode 100644 modules/eip/outputs.tf create mode 100644 modules/eip/variables.tf create mode 100644 modules/sg/main.tf create mode 100644 modules/sg/outputs.tf create mode 100644 modules/sg/variables.tf delete mode 100644 modules/vm_linux/main.tf create mode 100644 modules/vpc/main.tf create mode 100644 modules/vpc/outputs.tf create mode 100644 public_ip.txt</pre>
<p><b>git push origin dev</b></p>	<pre>\$ git push origin main Enumerating objects: 27, done. Counting objects: 100% (27/27), done. Delta compression using up to 4 threads Compressing objects: 100% (21/21), done. Writing objects: 100% (23/23), 4.18 KiB   118.00 KiB/s, done. Total 23 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0) remote: Resolving deltas: 100% (1/1), completed with 1 local object. To https://github.com/fabrice-git-hub/terraformcloud-aws-vm.git 4604871..6b72c1b main -&gt; main</pre>
<p><b>Github</b></p> <p><a href="https://github.com/fabrice-git-hub/terraformcloud-aws-vm/tree/main">https://github.com/fabrice-git-hub/terraformcloud-aws-vm/tree/main</a></p>	



Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>  <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 4. ANNEXE

### 4.1. MAINT.TF

#### a. MAINT.TF (racine)

```
terraform {
  You, 7 days ago | 1 author (You)
  required_providers {
    You, last week | 1 author (You)
    aws = {
      source  = "hashicorp/aws"
      version = "5.95.0" # version du provider AWS terraform
    }
    You, 7 days ago | 1 author (You)
    vault = {
      source  = "hashicorp/vault"
      version = "4.7.0" # version du provider VAULT terraform
    }
  }
  required_version = "1.11.4" # version du binaire terraform en local
}
```

```
You, 7 days ago | 1 author (You)
data "vault_aws_access_credentials" "creds" {
  backend = "aws"
  role    = "my-role"
}
```

```
You, 7 days ago | 1 author (You)
provider "aws" {
  region      = "us-east-1" # déclaration de la région
  access_key  = data.vault_aws_access_credentials.creds.access_key
  secret_key  = data.vault_aws_access_credentials.creds.secret_key
}
```

```
You, 7 days ago | 1 author (You)
data "vault_aws_access_credentials" "creds" {
  backend = "aws"
  role    = "my-role"
}
```

```
You, 7 days ago | 1 author (You)
provider "aws" {
  region      = "us-east-1" # déclaration de la région
  access_key  = data.vault_aws_access_credentials.creds.access_key
  secret_key  = data.vault_aws_access_credentials.creds.secret_key
}
```

```
You, 7 days ago | 1 author (You)
provider "vault" {
  address = "http://127.0.0.1:8200"
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
<b>ECF4</b>	Titre		Du	10/11/2025

```

module "ec2" {
  source = "../modules/ec2" # répertoire contenant le main.tf pour création de l'instance EC2
  security_group_id = module.sg.aws_security_group_id
  network_interface_id = module.eip.aws_eni_id
}

You, 2 hours ago | 1 author (You)
module "vpc" {
  source = "../modules/vpc" # répertoire contenant le main.tf pour création des instances VPC/Gateway/Route/Subnet
}

You, 2 hours ago | 1 author (You)
module "eip" {
  source = "../modules/eip" # répertoire contenant le main.tf pour création de l'instance EIP et ENI
  subnet_id = module.vpc.aws_subnet_id
  security_group_id = module.sg.aws_security_group_id
}

You, 2 hours ago | 1 author (You)
module "sg" {
  source = "../modules/sg" # répertoire contenant le main.tf pour création de l'instance SecurityGroup
  vpc_id = module.vpc.aws_vpc_id
}

resource "null_resource" "export_public_ip" {
  depends_on = [module.eip]
  You, 2 hours ago | 1 author (You)
  provisioner "local-exec" {
    command = "echo ${module.eip.public_ip} > public_ip.txt"
  }
}

```

## b. MAINT.TF (EC2)

```

resource "aws_instance" "debian_bullseye_1" {
  ami = "ami-000a08b963606bb82" # debian-11-amd64-20250402-2070 sur eu-east-1
  instance_type = "t2.micro" # taille de l'instance EC2 (1 vCPU/ 1 GB)
  key_name = "devops-fabrice" # Nom de la paire de clés SSH créée dans AWS.
  # security_groups = [var.security_group_id] # Associe groupe de sécurité à l'instance Debian

  network_interface {
    device_index = 0 # Spécifie l'index de l'interface réseau

    # L'instance Debian utilisera l'interface réseau network_interface_1
    network_interface_id = var.network_interface_id
  }

  user_data = file("${path.module}/apache_install.sh")

  tags = {
    Name = "DebianBullseye_1" # Identification de la ressource dans la console AWS
  }
}

```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

### c. MAINT.TF (EIP)

```
# Création Elastic IP (EIP)
resource "aws_eip" "eip_1" {
  domain = "vpc" # Permet l'attache à une ENI dans un VPC
  tags = {
    Name = "ElasticIP_1" # Identification de la ressource dans la console AWS
  }
}

# Création Network Interface (ENI)
resource "aws_network_interface" "network_interface_1" {
  subnet_id = var.subnet_id # Associe cette interface au sous-réseau subnet_1
  private_ips = ["192.168.1.20"] # Adresse IP privée spécifique
  security_groups = [var.security_group_id]
  tags = {
    Name = "NetworkInterface_1" # Identification de la ressource dans la console AWS
  }
}

# Attacher l'Elastic IP (EIP) à Network Interface (ENI)
resource "aws_eip_association" "eip_assoc_1" {
  allocation_id = aws_eip.eip_1.id # ID IP élastique (EIP) associée à l'interface réseau
  network_interface_id = aws_network_interface.network_interface_1.id # ID de l'interface réseau (ENI) associée à l'EIP
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

#### d. MAINT.TF (SG)

```

resource "aws_security_group" "security_group_1" {
  name           = "security-group-1"
  description    = "Security group to allow SSH, HTTP, and HTTPS"
  vpc_id        = var.vpc_id # Associe ce security group à vpc_1

  # Autoriser les connexions SSH (port 22)
  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Autoriser les connexions HTTP (port 80)
  ingress {
    from_port = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Autoriser les connexions HTTPS (port 443)
  ingress {
    from_port = 443
    to_port   = 443
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Autoriser le trafic sortant
  You, 43 minutes ago | 1 author (You)
  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1" # Autorise tout le trafic sortant
    cidr_blocks = ["0.0.0.0/0"]
  }
  You, 43 minutes ago | 1 author (You)
  tags = {
    Name = "SecurityGroup_1" # Identification de la ressource dans la console AWS
  }
}

```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b> <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

## e. MAINT.TF (VPC)

```
# Création VPC avec bloc CIDR spécifique
resource "aws_vpc" "vpc_1" {
  cidr_block      = "192.168.0.0/16" # Bloc CIDR utilisé avec le VPC
  enable_dns_support = true          # Active la prise en charge DNS dans le VPC
  enable_dns_hostnames = true        # Permet la création de noms d'hôtes DNS dans le VPC
  tags = {
    Name = "aws_vpc_1" # Identification de la ressource dans la console AWS
  }
}

# Création Internet Gateway relié au VPC
resource "aws_internet_gateway" "igw_1" {
  vpc_id = aws_vpc.vpc_1.id # Associe l'Internet Gateway à vpc_1
  tags = {
    Name = "InternetGateway_1" # Identification de la ressource dans la console AWS
  }
}

# Création Table de Routage
resource "aws_route_table" "route_table_1" {
  vpc_id = aws_vpc.vpc_1.id # Associe cette table de routage à vpc_1
  route {
    cidr_block = "0.0.0.0/0" # Cette route permet l'accès à Internet
    gateway_id = aws_internet_gateway.igw_1.id # Trafic destiné à l'extérieur du VPC envoyé à IGW
  }
  tags = {
    Name = "RouteTable_1" # Identification de la ressource dans la console AWS
  }
}

# Création Subnet avec bloc CIDR spécifique
resource "aws_subnet" "subnet_1" {
  vpc_id      = aws_vpc.vpc_1.id # Associe ce sous-réseau à vpc_1
  cidr_block  = "192.168.1.0/24" # Bloc CIDR du sous-réseau
  availability_zone = "us-east-1a" # Zone de disponibilité cible
  tags = {
    Name = "Subnet_1" # Identification de la ressource dans la console AWS
  }
}

# Association Table de Routage / Subnet
resource "aws_route_table_association" "route_table_association_1" {
  subnet_id      = aws_subnet.subnet_1.id # ID du sous-réseau 192.168.1.0/24
  route_table_id = aws_route_table.route_table_1.id # ID de la table de routage
}
```

Evaluation	Parcours	ADMINISTRATEUR SYSTEME DEVOPS	Auteur	FT
ECF4	Titre		Du	10/11/2025

## 4.2. Terraform PLAN

```
$ terraform plan
data.vault_aws_access_credentials.creds: Reading...
data.vault_aws_access_credentials.creds: Still reading... [10s elapsed]
data.vault_aws_access_credentials.creds: Still reading... [20s elapsed]
data.vault_aws_access_credentials.creds: Read complete after 22s [id=aws/creds/my-

Terraform used the selected providers to generate the following execution plan. Re
+ create

Terraform will perform the following actions:

# null_resource.export_public_ip will be created
+ resource "null_resource" "export_public_ip" {
+ id = (known after apply)
}

# module.ec2.aws_instance.debian_bullseye_1 will be created
+ resource "aws_instance" "debian_bullseye_1" {
+ ami = "ami-000a08b963606bb82"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = "devops-fabrice"
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns = (known after apply)
+ private_ip = (known after apply)

+ public_dns = (known after apply)
+ public_ip = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups = (known after apply)
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags = {
+ "Name" = "DebianBullseye_1"
}
+ tags_all = {
+ "Name" = "DebianBullseye_1"
}
+ tenancy = (known after apply)
+ user_data = "18e971c1eb8b89f2139e768e173ce"
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)

+ capacity_reservation_specification (known after apply)

+ cpu_options (known after apply)

+ ebs_block_device (known after apply)

+ enclave_options (known after apply)

+ ephemeral_block_device (known after apply)

+ instance_market_options (known after apply)

+ maintenance_options (known after apply)

+ metadata_options (known after apply)

+ network_interface {
+ delete_on_termination = false
+ device_index = 0
+ network_card_index = 0
+ network_interface_id = (known after apply)
}

+ private_dns_name_options (known after apply)

+ root_block_device (known after apply)
}
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>	Auteur	FT
ECF4	Titre	Mettre l'Infrastructure en Production dans le Cloud	Du	10/11/2025

```
# module.eip.aws_eip.eip_1 will be created
+ resource "aws_eip" "eip_1" {
  + allocation_id      = (known after apply)
  + arn                = (known after apply)
  + association_id     = (known after apply)
  + carrier_ip         = (known after apply)
  + customer_owned_ip  = (known after apply)
  + domain             = "vpc"
  + id                 = (known after apply)
  + instance           = (known after apply)
  + ipam_pool_id       = (known after apply)
  + network_border_group = (known after apply)
  + network_interface   = (known after apply)
  + private_dns        = (known after apply)
  + private_ip         = (known after apply)
  + ptr_record         = (known after apply)
  + public_dns         = (known after apply)
  + public_ip          = (known after apply)
  + public_ipv4_pool    = (known after apply)
  + tags               = {
    + "Name" = "ElasticIP_1"
  }
  + tags_all           = {
    + "Name" = "ElasticIP_1"
  }
  + vpc                = (known after apply)
}

# module.eip.aws_eip_association.eip_assoc_1 will be created
+ resource "aws_eip_association" "eip_assoc_1" {
  + allocation_id      = (known after apply)
  + id                 = (known after apply)
  + instance_id        = (known after apply)
  + network_interface_id = (known after apply)
  + private_ip_address  = (known after apply)
  + public_ip          = (known after apply)
}
```

```
# module.eip.aws_network_interface.network_interface_1 will be created
+ resource "aws_network_interface" "network_interface_1" {
  + arn                = (known after apply)
  + enable_primary_ipv6 = (known after apply)
  + id                 = (known after apply)
  + interface_type     = (known after apply)
  + ipv4_prefix_count  = (known after apply)
  + ipv4_prefixes      = (known after apply)
  + ipv6_address_count = (known after apply)
  + ipv6_address_list  = (known after apply)
  + ipv6_address_list_enabled = false
  + ipv6_addresses     = (known after apply)
  + ipv6_prefix_count  = (known after apply)
  + ipv6_prefixes      = (known after apply)
  + mac_address        = (known after apply)
  + outpost_arn        = (known after apply)
  + owner_id           = (known after apply)
  + private_dns_name    = (known after apply)
  + private_ip         = (known after apply)
  + private_ip_list    = (known after apply)
  + private_ip_list_enabled = false
  + private_ips        = [
    + "192.168.1.20",
  ]
  + private_ips_count  = (known after apply)
  + security_groups     = (known after apply)
  + source_dest_check   = true
  + subnet_id           = (known after apply)
  + tags               = {
    + "Name" = "NetworkInterface_1"
  }
  + tags_all           = {
    + "Name" = "NetworkInterface_1"
  }
}
+ attachment (known after apply)
}
```



Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>  <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

```
# module.sg.aws_security_group.security_group_1 will be created
+ resource "aws_security_group" "security_group_1" {
+   arn                = (known after apply)
+   description        = "Security group to allow SSH, HTTP, and HTTPS"
+   egress              = [
+     {
+       cidr_blocks     = [
+         "0.0.0.0/0",
+       ]
+       from_port        = 0
+       ipv6_cidr_blocks = []
+       prefix_list_ids  = []
+       protocol         = "-1"
+       security_groups  = []
+       self              = false
+       to_port          = 0
+     },
+     {
+       cidr_blocks     = [
+         "0.0.0.0/0",
+       ]
+       from_port        = 22
+       ipv6_cidr_blocks = []
+       prefix_list_ids  = []
+       protocol         = "tcp"
+       security_groups  = []
+       self              = false
+       to_port          = 22
+     },
+     {
+       cidr_blocks     = [
+         "0.0.0.0/0",
+       ]
+       from_port        = 443
+       ipv6_cidr_blocks = []
+       prefix_list_ids  = []
+       protocol         = "tcp"
+       security_groups  = []
+       self              = false
+       to_port          = 443
+     },
+   ],
+   id                = (known after apply)
+   ingress            = [
+     {
+       cidr_blocks     = [
+         "0.0.0.0/0",
+       ]
+       from_port        = 80
+       ipv6_cidr_blocks = []
+       prefix_list_ids  = []
+       protocol         = "tcp"
+       security_groups  = []
+       self              = false
+       to_port          = 80
+     },
+   ],
+   name               = "security-group-1"
+   name_prefix        = (known after apply)
+   owner_id            = (known after apply)
+   revoke_rules_on_delete = false
+   tags                = {
+     "Name" = "SecurityGroup_1"
+   }
+   tags_all            = {
+     "Name" = "SecurityGroup_1"
+   }
+   vpc_id              = (known after apply)
+ }

# module.vpc.aws_internet_gateway.igw_1 will be created
+ resource "aws_internet_gateway" "igw_1" {
+   arn                = (known after apply)
+   id                 = (known after apply)
+   owner_id            = (known after apply)
+   tags                = {
+     "Name" = "InternetGateway_1"
+   }
+   tags_all            = {
+     "Name" = "InternetGateway_1"
+   }
+   vpc_id              = (known after apply)
+ }
```

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>  <b>Mettre l'Infrastructure en Production dans le Cloud</b>	Auteur	FT
ECF4	Titre		Du	10/11/2025

```
# module.vpc.aws_route_table.route_table_1 will be created
+ resource "aws_route_table" "route_table_1" {
+   arn                = (known after apply)
+   id                 = (known after apply)
+   owner_id           = (known after apply)
+   propagating_vgws   = (known after apply)
+   route              = [
+     + {
+       + cidr_block          = "0.0.0.0/0"
+       + gateway_id         = (known after apply)
+       # (11 unchanged attributes hidden)
+     },
+   ]
+   tags               = {
+     + "Name" = "RouteTable_1"
+   }
+   tags_all           = {
+     + "Name" = "RouteTable_1"
+   }
+   vpc_id             = (known after apply)
+ }

# module.vpc.aws_route_table_association.route_table_association_1 will be created
+ resource "aws_route_table_association" "route_table_association_1" {
+   id                 = (known after apply)
+   route_table_id     = (known after apply)
+   subnet_id          = (known after apply)
+ }
```

```
# module.vpc.aws_vpc.vpc_1 will be created
+ resource "aws_vpc" "vpc_1" {
+   arn                = (known after apply)
+   cidr_block         = "192.168.0.0/16"
+   default_network_acl_id = (known after apply)
+   default_route_table_id = (known after apply)
+   default_security_group_id = (known after apply)
+   dhcp_options_id     = (known after apply)
+   enable_dns_hostnames = true
+   enable_dns_support   = true
+   enable_network_address_usage_metrics = (known after apply)
+   id                  = (known after apply)
+   instance_tenancy    = "default"
+   ipv6_association_id = (known after apply)
+   ipv6_cidr_block      = (known after apply)
+   ipv6_cidr_block_network_border_group = (known after apply)
+   main_route_table_id = (known after apply)
+   owner_id            = (known after apply)
+   tags               = {
+     + "Name" = "aws_vpc_1"
+   }
+   tags_all           = {
+     + "Name" = "aws_vpc_1"
+   }
+ }
```

**Plan:** 11 to add, 0 to change, 0 to destroy.

Evaluation	Parcours	<b>ADMINISTRATEUR SYSTEME DEVOPS</b>  Mettre l'Infrastructure en Production dans le Cloud	Auteur	FT
ECF4	Titre		Du	10/11/2025

### 4.3. Troubleshooting

## Appel d'un Module dans un Module

### Error: Reference to undeclared module

```
on modules\sg\main.tf line 4, in resource "aws_security_group" "security_group_1":
4:   vpc_id      = module.vpc.aws_vpc_id # Associe ce security group à vpc_1

No module call named "vpc" is declared in module.sg.
```

**Il n'est pas conseillé d'appeler un module directement d'un autre module mais de passer par les variables.**

Si on doit par exemple faire appel au module VPC dans le module SG, il faut passer l'ID du VPC comme une variable depuis le module racine (le main.tf principal).

1. Dans **modules/sg/variables.tf**, on ajoute

```
variable "vpc_id" {
  type = string
}
```

2. Dans **modules/sg/main.tf**, on modifie la ressource

```
vpc_id = var.vpc_id
```

3. Dans le fichier principal (main.tf à la racine), quand on appelle le module SG, on ajoute l'Output du module VPC nommé aws\_vpc\_id qui va permettre d'associer le Security Group au VPC

```
module "sg" {
  source = "../modules/sg" # répertoire contenant le main.tf pour création de
  l'instance SecurityGroup
  vpc_id = module.vpc.aws_vpc_id ←
}
```