

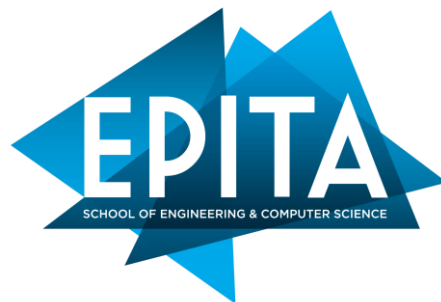
PROJET TERRAFORM OUTSCALE

Vi Long LUONG <vi-long.luong@epita.fr>

Uriel SAMBANI <sambiani.uriel@epita.fr>

CYB_3A

DS OUTSCALE



1. Présentation de l'Architecture	3
1.1 Objectif.....	3
1.2 Schéma de l'Architecture	3
1.3 Composants Clés.....	3
2. Instructions Pas-à-pas.....	5
2.1 Initialiser et Appliquer le Projet Terraform	5
2.2 Se connecter à la Machine Bastion.....	7
2.3 Vérifier que l'Application Web est en Ligne.....	8
2.4 Détruire les ressources créées:	10
3. Concepts Clés	11
3.1 NAT Gateway.....	11
3.2 Machine Bastion	11
3.3 Load Balancer.....	11
4. Annexes	12
4.1 Commandes Terraform Utiles	12
5. Conclusion.....	13

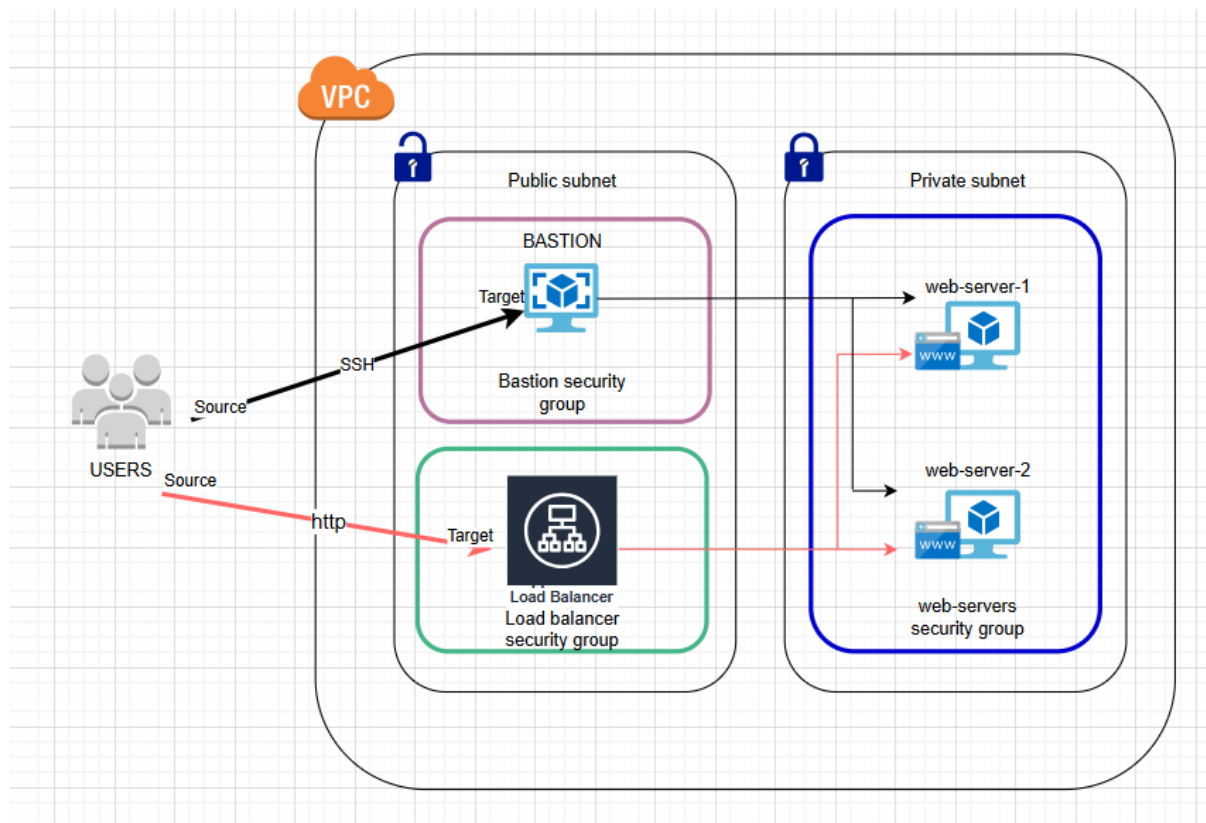
1. Présentation de l'Architecture

1.1 Objectif

L'architecture déployée permet de :

- Créer un **réseau sécurisé** avec un sous-réseau public et un sous-réseau privé.
- Utiliser une **machine bastion** pour accéder aux serveurs situés dans le sous-réseau privé.
- Déployer deux **serveurs web** dans le sous-réseau privé avec une application web.
- Configurer un **Load Balancer** pour distribuer le trafic entre les serveurs web.

1.2 Schéma de l'Architecture



1.3 Composants Clés

1. **VPC et Sous-réseaux :**
 - **Sous-réseau public :** Héberge la machine Bastion et le Load Balancer.

- **Sous-réseau privé** : Héberge deux serveurs web accessibles uniquement en SSH via la Bastion.
- 2. **Machine Bastion** :
 - Point d'accès **SSH** sécurisé vers les deux machines privées.
 - Accessible uniquement via le port 22 (SSH).
- 3. **Serveurs Web** :
 - Hébergent une application web simple.
 - L'application est configurée automatiquement via **Cloud-init** lors de la création des machines.
- 4. **Load Balancer** :
 - Répartit le trafic entrant vers les deux serveurs web.
 - Vérifie la **santé des serveurs** avant d'envoyer des requêtes.
- 5. **NAT Gateway** :
 - Permet aux serveurs situés dans le sous-réseau privé d'accéder à Internet pour les mises à jour (sans être accessibles depuis l'extérieur).

2. Instructions Pas-à-pas

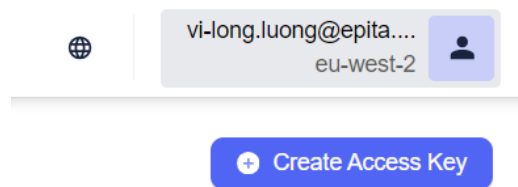
2.1 Initialiser et Appliquer le Projet Terraform

1. Prérequis :

- Terraform installé sur votre machine.
- Accès à un compte Outscale avec des clés d'accès configurées.

2. Configuration d'accès:

- Créer une nouvelle clé d'accès sur la plateforme **Outscale** (Ne pas oublier d'enregistrer **Access_key_ID** et **Secret_Key_ID**)



- Remplacer les deux variables définies dans le fichier **terraform.tfvars** avec votre clé d'accès

```
access_key_id = "<Votre access_key_id>"
secret_key_id = "<Votre secret_key_id>"
```

Structure des fichiers :

Assurez-vous d'avoir les fichiers suivants dans votre répertoire de projet Terraform :

```
project/
├── bastion.tf
├── keypair.tf
├── load_balancer.tf
├── main.tf
├── network.tf
├── security_group.tf
├── terraform.tfvars
├── variables.tf
├── web_servers.tf
└── cloud-init-web-server.yml
```

3. Initialisation du projet :

Ouvrez un terminal dans le dossier du projet et exécutez :

terraform init

```
vi-long-luong@epita:~/CloudSouverain/OUTSCALE/outscaleproject$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding outscale/outscale versions matching ">= 0.11.0"...
- Finding latest version of hashicorp/tls...
- Finding latest version of hashicorp/local...
- Installing hashicorp/tls v4.0.6...
- Installed hashicorp/tls v4.0.6 (signed by HashiCorp)
- Installing hashicorp/local v2.5.2...
- Installed hashicorp/local v2.5.2 (signed by HashiCorp)
- Installing outscale/outscale v0.12.0...
- Installed outscale/outscale v0.12.0 (signed by a HashiCorp partner, key ID E88D052B326BFCEE)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

4. Vérification des changements :

terraform plan

```
vi-long-luong@epita:~/CloudSouverain/OUTSCALE/outscaleproject$ terraform plan

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# local_file.my_key will be created
+ resource "local_file" "my_key" {
  + content             = (sensitive value)
  + content_base64sha256 = (known after apply)
  + content_base64sha512 = (known after apply)
  + content_md5         = (known after apply)
  + content_sha1        = (known after apply)
  + content_sha256      = (known after apply)
  + content_sha512      = (known after apply)
  + directory_permission = "0777"
  + file_permission     = "0600"
  + filename            = "./my_key.pem"
  + id                  = (known after apply)
}

# outscale_internet_service.internet_service_as will be created
+ resource "outscale_internet_service" "internet_service_as" {
  + id              = (known after apply)
  + internet_service_id = (known after apply)
  + net_id          = (known after apply)
  + request_id      = (known after apply)
  + state           = (known after apply)
}

# outscale_internet_service_link.igw_link will be created
+ resource "outscale_internet_service_link" "igw_link" {
  + id              = (known after apply)
  + internet_service_id = (known after apply)
  + net_id          = (known after apply)
  + request_id      = (known after apply)
  + state           = (known after apply)
  + tags            = (known after apply)
}

# outscale_keypair.my_keypair will be created
+ resource "outscale_keypair" "my_keypair" {
  + id              = (known after apply)
  + keypair_fingerprint = (known after apply)
  + keypair_name      = "outscale_terraform_project"
  + private_key       = (known after apply)
  + public_key        = (known after apply)
  + request_id       = (known after apply)
}
```

5. Appliquer la configuration :

Lancez le déploiement de l'architecture :

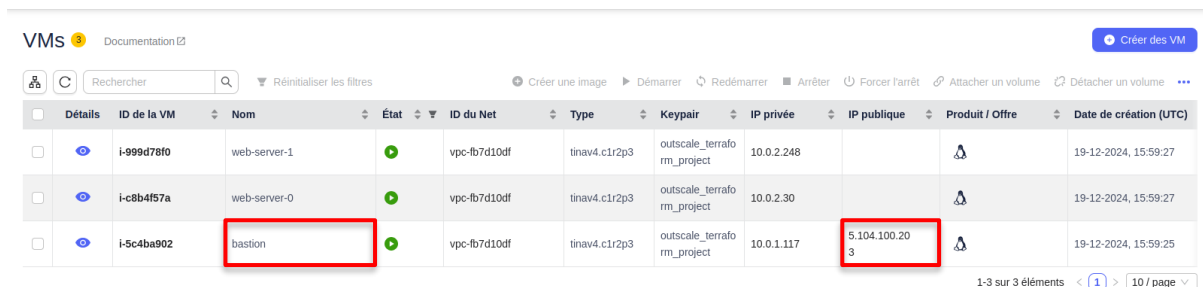
```
terraform apply
```

Après le déploiement, vous trouverez sur la plateforme **Outscale** les ressources créées par Terraform.

2.2 Se connecter à la Machine Bastion

1. Obtenez l'adresse IP publique de la bastion :

Allez sur l'onglet **Compute > VMs** sur la plateforme **Outscale** et cherchez la VM qui s'appelle **"bastion"**. Copiez son **Public IP**



	Détails	ID de la VM	Nom	État	ID du Net	Type	Keypair	IP privée	IP publique	Produit / Offre	Date de création (UTC)
<input type="checkbox"/>		i-999d78f0	web-server-1		vpc-fb7d10df	tinav4.c1r2p3	outscale_terraform_project	10.0.2.248			19-12-2024, 15:59:27
<input type="checkbox"/>		i-c8b4f57a	web-server-0		vpc-fb7d10df	tinav4.c1r2p3	outscale_terraform_project	10.0.2.30			19-12-2024, 15:59:27
<input type="checkbox"/>		i-5c4ba902	bastion		vpc-fb7d10df	tinav4.c1r2p3	outscale_terraform_project	10.0.1.117	5.104.100.203		19-12-2024, 15:59:25

Utilisez une clé SSH pour vous connecter : En utilisant la clé privée SSH **my_key.pem** créée pendant le déploiement, exécutez les commandes suivantes depuis un terminal :

```
chmod 400 my_key.pem
```

```
ssh -i my_key.pem outscale@<IP-PUBLIQUE-BASTION>
```

```
vi-long-luong@epita:~/CloudSouverain/OUTSCALE/outscaleproject$ ssh -i my_key.pem outscale@5.104.100.203
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-56-generic x86_64)

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

System Information as of Thu Dec 19 17:28:00 UTC 2024

System load:  0.0               Processes:    140
Usage of /:   16.7% of 9.51GB   Users logged in:  0
Memory usage: 10%              IPv4 address for eth0: 10.0.1.117
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

2 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

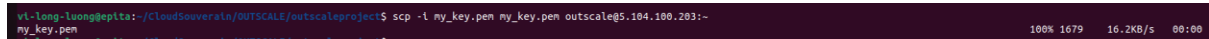
Last login: Thu Dec 19 16:36:16 2024 from 82.65.101.59
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

outscale@ip-10-0-1-117:~$
```

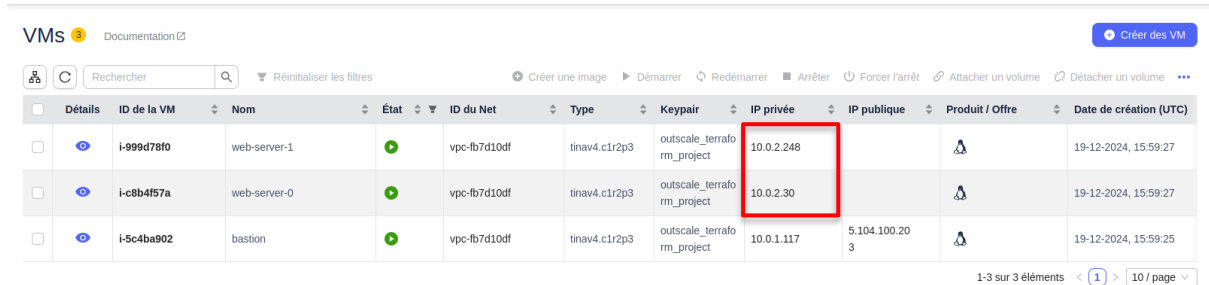
2. Accès aux serveurs web via la bastion :

Envoyez la clé privée **my_key.pem** à la machine **bastion** en utilisant **scp**

```
scp -i my_key.pem my_key.pem outscale@<IP-PUBLIQUE-BASTION>:~
```



Dans l'onglet **Compute > VMs**, vous trouverez les **Private IP** des deux machines privées.

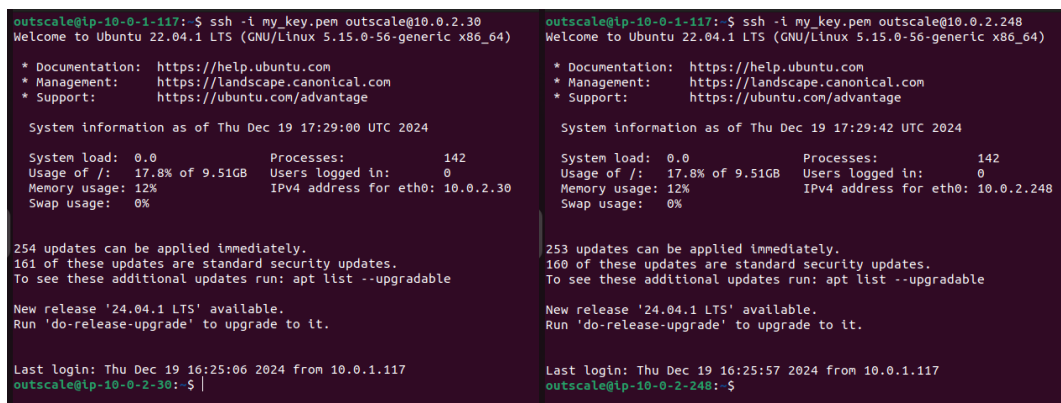


	Détails	ID de la VM	Nom	État	ID du Net	Type	Keypair	IP privée	IP publique	Produit / Offre	Date de création (UTC)
<input type="checkbox"/>		i-999d78f0	web-server-1		vpc-lb7d10df	tinav4.c1r2p3	outscale_terraform_project	10.0.2.248			19-12-2024, 15:59:27
<input type="checkbox"/>		i-c8b4f57a	web-server-0		vpc-lb7d10df	tinav4.c1r2p3	outscale_terraform_project	10.0.2.30			19-12-2024, 15:59:27
<input type="checkbox"/>		i-5c4ba902	bastion		vpc-lb7d10df	tinav4.c1r2p3	outscale_terraform_project	10.0.1.117	5.104.100.203		19-12-2024, 15:59:25

Depuis la machine **bastion**, connectez-vous aux serveurs web privés :

```
ssh -i my_key.pem outscale@<IP-PRIVEE-SERVER1>
```

```
ssh -i my_key.pem outscale@<IP-PRIVEE-SERVER2>
```



2.3 Vérifier que l'Application Web est en Ligne

1. Obtenez l'adresse IP ou DNS du Load Balancer :

Allez sur l'onglet **Load Balancing Unit > Load Balancers** et copiez **Public IP** du Load Balancer.

Load Balancers ⓘ Documentation ⓘ										Créer un load balancer
Rechercher										Modifier Supprimer
	Détails	Nom	Type	ID du Net	IP publique	Nom du DNS	Security groups	Sous-régi...	Tags	
		public-load-balancer	internet-facing	vpc-fb7d10df	109.232.238.245	public-load-balancer-064944556.eu-west-2.lb.amazonaws.com	sg-75368db0	eu-west-2a	name: web-b	

Accédez à l'application via un navigateur :

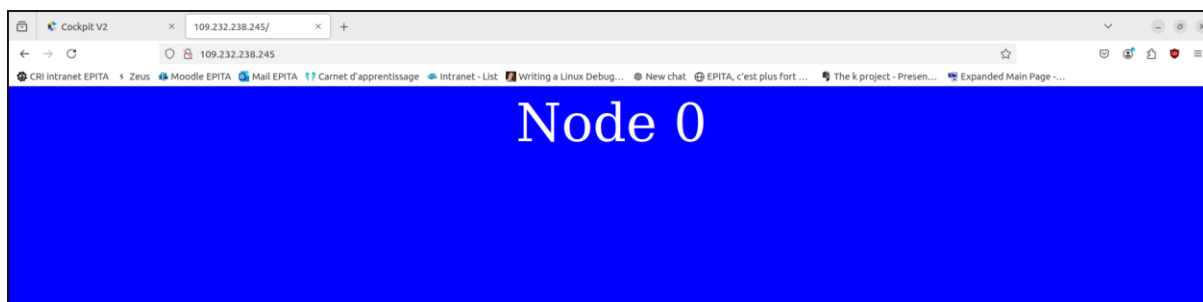
`http://<IP-LOAD-BALANCER>`

Une autre manière d'y accéder est de cliquer directement sur le **Nom du DNS** du Load Balancer

Load Balancers ⓘ Documentation ⓘ										Créer un load balancer
Rechercher										Modifier Supprimer
	Détails	Nom	Type	ID du Net	IP publique	Nom du DNS	Security groups	Sous-régi...	Tags	
		public-load-balancer	internet-facing	vpc-fb7d10df	109.232.238.245	public-load-balancer-064944556.eu-west-2.lb.amazonaws.com	sg-75368db0	eu-west-2a	name: web-b	

2. Vérification :

- Vous devriez voir la page web avec un message indiquant le **nom du serveur**.
- Le **Load Balancer** distribue le trafic entre les deux serveurs (actualisez la page plusieurs fois pour voir l'alternance des serveurs).



2.4 Détruire les ressources créées:

Pour détruire les ressources créées sur la plateforme **Outscale** avec **Terraform**, il suffit de lancer la commande suivante:

```
terraform destroy
```

3. Concepts Clés

3.1 NAT Gateway

- Un **NAT Gateway** permet aux instances situées dans un sous-réseau privé d'accéder à Internet pour, par exemple, effectuer des mises à jour ou télécharger des dépendances.
 - Les ressources du sous-réseau privé ne sont **pas accessibles directement depuis Internet** pour des raisons de sécurité.
-

3.2 Machine Bastion

- Une **machine Bastion** est une instance située dans un sous-réseau public.
 - Elle sert de point d'accès sécurisé **SSH** pour se connecter aux ressources situées dans un sous-réseau privé.
 - La bastion limite les accès directs aux serveurs privés pour renforcer la sécurité.
-

3.3 Load Balancer

- Un **Load Balancer** répartit le trafic réseau entrant entre plusieurs serveurs pour améliorer la disponibilité et la tolérance aux pannes.
- Il vérifie la santé des serveurs avant de leur envoyer du trafic, garantissant ainsi un service continu.
- Dans cette architecture, le Load Balancer distribue le trafic entre deux serveurs web.

4. Annexes

4.1 Commandes Terraform Utiles

- Initialiser le projet :

```
terraform init
```

- Vérifier les modifications qui seront appliquées :

```
terraform plan
```

- Appliquer les changements :

```
terraform apply
```

- Détruire les ressources créées :

```
terraform destroy
```

5. Conclusion

Cette documentation fournit les étapes nécessaires pour déployer, utiliser et vérifier une architecture réseau et applicative basée sur **Terraform** et **Cloud-Init**.

L'architecture garantit une **sécurité renforcée** grâce à l'utilisation de **sous-réseau privé**, d'une machine **Bastion** pour sécuriser les accès aux ressources sensibles et d'un **Load Balancer** pour la répartition du trafic. Elle met en place aussi un **service NAT** qui permet à des instances du sous-réseau privé d'accéder à l'Internet afin d'effectuer des mises à jour ou des téléchargements de dépendances.

La solution fournie est robuste mais reste simple à comprendre et facile à gérer, ce qui diminue le temps d'adaptation pour les utilisateurs, même non-spécialiste, et facilite la maintenance dans l'avenir.