

# Démarrage avec Terraform et Scaleway

Tutoriel de prise en main de Scaleway avec Terraform.

La fourniture des credential Scaleway sera effectué avec la CLI SCW. Cela évitera tout leak de credential.

## Installation de Terraform

La documentation Terraform est plus adapté car la méthode d'installation dépend fortement de votre environnement d'exécution Documentation

## CLI Scaleway

### Installation de CLI Scaleway

```
sudo curl -o /usr/local/bin/scw -L "https://github.com/scaleway/scaleway-cli/releases/download/v2.4.0/scw-"
```

```
sudo chmod +x /usr/local/bin/scw
```

### Configuration de la CLI SCW

```
scw init
```

Enter a valid secret-key or an email: <votre\_mail>

Enter your password: <votre mot de passe il ne s'affichera pas c'est normal>

To improve this tool we rely on diagnostic and usage data.

Sending such data is optional and can be disabled at any time by running "scw config set send-telemetry=false"

Do you want to send usage statistics and diagnostics? (Y/n): y

To fully enjoy Scaleway CLI we recommend you install autocomplete support in your shell.

Do you want to install autocomplete? (Y/n): y

Creating new config at /home/bbalaud/.config/scw/config.yaml

Config saved at /home/bbalaud/.config/scw/config.yaml:

access\_key: SCWA7Z3HSQ5T46GT9EOV

secret\_key: 26acf066-xxxx-xxxx-xxxx-xxxxxxxxxxxx

default\_organization\_id: 47fd9bd7-e93b-40e9-98fe-3691351fa3f4

default\_project\_id: 47fd9bd7-e93b-40e9-98fe-3691351fa3f4

default\_region: fr-par

default\_zone: fr-par-1

To enable autocomplete, scw needs to update your shell configuration.

What type of shell are you using (default: bash):

To enable autocomplete we need to append to ~/.bashrc the following lines:

```
# Scaleway CLI autocomplete initialization.  
eval "$(scw autocomplete script shell=bash)"
```

Do you want to proceed with these changes? (Y/n): y

[V] Initialization completed with success.

Except for SSH key: could not find an SSH key at ~/.ssh/id\_rsa.pub

## Premier Terraform

Création du répertoire pour stocker le projet Terraform

```
mkdir mon_super_terraform
cd mon_super_terraform
```

Création du premier fichier Terraform. Les fichiers Terraform doivent forcément se terminer par l'extension .tf.

```
vim main.tf
```

Contenu de main.tf

```
terraform {
  required_providers {
    scaleway = {
      source = "scaleway/scaleway"
    }
  }
  required_version = ">= 0.13"
}

provider "scaleway" {
  zone   = "fr-par-1"
  region = "fr-par"
}

resource "scaleway_instance_ip" "public_ip" {
}

resource "scaleway_instance_server" "web" {
  type      = "DEV1-S"
  image     = "ubuntu_focal"
  tags     = ["front", "web"]
  ip_id    = scaleway_instance_ip.public_ip.id
  root_volume {}
}
```

Initialisation de Terraform, Planification et application.

```
terraform init
terraform plan
terraform apply
```

Une fois vos tests effectués vous pouvez détruire les ressources avec :

```
terraform destroy
```

## Terraform avec une organisation secondaire

Les clés d'API Scaleway sont liées à une organisation. Autrement dit vos clés d'API générées sur votre organisation personnelle ne peuvent pas intervenir sur une autre organisation.

Il faut donc générer une nouvelle clé d'API dans l'organisation sur laquelle vous souhaitez travailler. Cela est possible depuis la page "Project" dans l'onglet "Credential".

Veillez conservez les credentials de manière sécurisé et privée (donc pas dans git). Il ne vont serront pas refourni en cas de perte. Une fois cela fait vous pourrez modifier les credentials utilisé par Terraform via des variables d'environnements. (Vous devrez ré-exporter les variables d'environnements a chaque ouverture de sessions

```
export SCW_ACCESS_KEY="<YOUR ACCES KEY>"
export SCW_SECRET_KEY="<YOUR SECRET>"
```

Une fois les variables exporter, vous devez modifier la configuration du provider scaleway dans Terraform et ajouté la ligne project\_id avec le project\_id dans lequel vous souhaitez déployer :

```
provider "scaleway" {
  zone   = "fr-par-1"
  region = "fr-par"
  project_id = "<project_id>"
}
```

Maintenant avec terraform vous pouvez déployer sur autre projet.

## Git Ignore

Le fichier gitignore permet d'indiquer des pattern de fichier à ignorer. Cela permet déviter des pousser des fichiers non pertinents sur un dépôt git et/ou des fichiers contenant des informations sensibles.

```
curl https://raw.githubusercontent.com/github/gitignore/main/Terraform.gitignore --output .gitignore
git add .gitignore
git commit -m "add gitignore"
```

## Documentation utile

Ajouter un membre à une organisation scaleway : <https://www.scaleway.com/en/docs/console/my-project/how-to/manage-organization-members/#how-to-invite-members>

Utilisé Terraform et Ansible : <https://www.digitalocean.com/community/tutorials/how-to-use-ansible-with-terraform-for-configuration-management>

Ajouter une clés SSH : <https://www.scaleway.com/en/docs/console/my-project/how-to/create-ssh-key/#how-to-upload-the-public-ssh-key-to-the-scaleway-interface>

Documentation Terraform Scaleway : <https://registry.terraform.io/providers/scaleway/scaleway/latest/docs>