Orchestration d’infrastructure : les fondamentaux de Terraform

Prix : 750 €HT

Durée : 1 jour

Code de Référence : TERRAFON

Catalogue Cloud Native et DevOps

L’Infrastructure as Code est devenue une pratique centrale pour automatiser, fiabiliser et industrialiser le déploiement des infrastructures Cloud. Terraform permet de décrire une infrastructure sous forme de code, de versionner les changements et de déployer des ressources de manière reproductible.

Cette formation d’une journée présente les fondamentaux de Terraform, son positionnement dans l’écosystème DevOps et Cloud, ainsi que les principaux objets utilisés pour créer et organiser une infrastructure. Le format condensé privilégie les notions essentielles, les démonstrations guidées et la mise en pratique sur des cas simples. L’objectif est de permettre aux participants de comprendre rapidement la logique Terraform et de disposer des bases nécessaires pour lire, modifier et exécuter une configuration simple.

Objectifs de la formation

A l’issue de cette formation, vous serez capable de :

- Décrire les principes de l’Infrastructure as Code et les apports de Terraform.
- Identifier les principaux cas d’usage de Terraform dans un contexte Cloud ou infrastructure virtuelle.
- Installer et initialiser un projet Terraform simple.
- Expliquer le rôle des providers, ressources, variables, outputs et data sources.
- Écrire et exécuter une configuration Terraform élémentaire.
- Utiliser les commandes de base du cycle Terraform : init, plan, apply et destroy.

- Organiser un projet Terraform simple en respectant les premières bonnes pratiques.

- Identifier les notions à approfondir pour aller vers des usages avancés : modules, state, backend, multi-environnements et écosystème Terraform.

Public

Cette formation s'adresse aux

- Administrateurs systèmes souhaitant automatiser le déploiement d'infrastructures,
- Développeurs amenés à intervenir sur des environnements Cloud ou DevOps,
- Architectes techniques ou Cloud souhaitant comprendre les fondamentaux de Terraform,
- Ingénieurs DevOps débutant avec l'Infrastructure as Code,
- Chefs de projets techniques souhaitant mieux comprendre les principes d'orchestration d'infrastructure.

Prérequis

Avoir des connaissances de base en administration Linux / Unix.
Connaître les principes généraux du Cloud Computing.
Avoir des notions de ligne de commande.
Une première expérience en scripting, par exemple Bash, Python ou équivalent, est un plus.
Une connaissance de base d'un Cloud Provider, par exemple AWS, Azure ou Google Cloud, est un plus.

Programme de la formation

Les démonstrations, exemples et travaux pratiques sont donnés à titre indicatif et pourront être adaptés selon les besoins et le contexte de la session.

Jour 1
Matinée – Fondamentaux de l'Infrastructure as Code et premières manipulations Terraform
Introduction – Positionnement et cadrage de la journée

- Présentation des objectifs et du déroulé
- Tour de table des participants
- Rappel des notions utiles : Cloud, automatisation, ligne de commande, infrastructure virtuelle
- Présentation du rôle de Terraform dans une démarche DevOps et Infrastructure as Code

Chapitre 1 – Comprendre l'Infrastructure as Code et le positionnement de Terraform

- Définition de l'Infrastructure as Code
- Différence entre infrastructure manuelle, scripts ad hoc, gestion de configuration et orchestration d'infrastructure
- Notion d'infrastructure immutable
- Apports de l'IaC : reproductibilité, traçabilité, versionnement, automatisation, réduction des erreurs
- Positionnement de Terraform dans l'écosystème Cloud et DevOps

- Cas d’usage principaux de Terraform : création de ressources Cloud, environnements de test, socles techniques, automatisation d’infrastructure
- Présentation du langage HCL, HashiCorp Configuration Language

Exemples et démonstrations :

- Comparaison entre une création manuelle de ressource Cloud et une création déclarative avec Terraform
- Lecture commentée d’un premier fichier Terraform simple
- Présentation du cycle déclaratif : configuration, planification, application, destruction

Chapitre 2 – Installer Terraform et manipuler les commandes de base

- Présentation des modes d’installation de Terraform
- Vérification de l’installation et de la version
- Structure minimale d’un projet Terraform
- Initialisation d’un projet avec terraform init
- Validation et formatage d’une configuration
- Prévisualisation des changements avec terraform plan
- Application d’une configuration avec terraform apply
- Suppression des ressources avec terraform destroy
- Comprendre le principe du fichier d’état Terraform

Exemples de travaux pratiques :

- Installation ou vérification de Terraform sur le poste de travail
- Création d’un premier projet Terraform local
- Exécution des commandes init, validate, fmt, plan, apply et destroy
- Observation du fichier d’état généré par Terraform

Après-midi – Configuration Terraform, bonnes pratiques et ouverture vers les usages avancés
Chapitre 3 – Décrire une infrastructure simple avec Terraform

- Organisation des fichiers Terraform
- Rôle du fichier main.tf
- Déclaration des providers
- Déclaration des ressources
- Gestion des identifiants et bonnes pratiques de sécurité élémentaires
- Utilisation des variables
- Utilisation des fichiers de variables
- Utilisation des outputs
- Introduction aux data sources
- Introduction aux templates

Exemples et démonstrations :

- Lecture commentée d’une configuration utilisant un provider
- Paramétrage simple d’une ressource avec des variables
- Affichage d’informations utiles avec les outputs
- Présentation d’une data source pour récupérer une information existante

Exemples de travaux pratiques :

- Création ou adaptation d'une configuration Terraform simple
- Ajout de variables et d'outputs
- Exécution d'un plan pour vérifier les modifications prévues
- Correction guidée des erreurs courantes de syntaxe ou de configuration

Chapitre 4 – Organiser, réutiliser et sécuriser les bases d'un projet Terraform

- Introduction aux modules Terraform
- Intérêt de la modularité pour réutiliser du code
- Présentation du Terraform Registry
- Bonnes pratiques de nommage et d'organisation des fichiers
- Bonnes pratiques sur les variables et les outputs
- Premières règles de sécurité : ne pas stocker de secrets dans le code, utiliser les variables d'environnement, séparer configuration et secrets
- Introduction au state et aux backends distants
- Aperçu des usages avancés : boucles, conditions, meta-arguments, fonctions, multi-environnements, multi-cloud
- Aperçu de l'écosystème : TFLint, terraform-docs, Terraform Graph, Terragrunt, Infracost, Vault, tests d'infrastructure

Exemples et démonstrations :

- Lecture d'un module simple
- Recherche d'un module public dans le Terraform Registry
- Visualisation du principe de dépendance entre ressources
- Démonstration commentée d'une organisation de projet plus maintenable

Exemples de travaux pratiques :

- Réorganisation d'un projet Terraform simple en plusieurs fichiers
- Ajout d'un output ou d'une variable supplémentaire
- Analyse guidée d'un module existant
- Identification collective des améliorations possibles sur un projet Terraform fourni

Conclusion – Synthèse et plan d'action individuel

- Synthèse des notions clés de la journée
- Rappel du cycle de travail Terraform.
- Rappel des objets fondamentaux : providers, ressources, variables, outputs, data sources, modules
- Présentation des pistes d'approfondissement : state distant, modules avancés, CI/CD, sécurité, tests, multi-environnements
- Questions / réponses
- Auto-évaluation finale

Méthodes pédagogiques

- Accompagnement théorique et pratique avec une pédagogie alliant théorie, démonstrations et exercices guidés, puis mise en pratique des notions abordées avec une évaluation des manipulations réalisées.

- Le formateur tient compte de la situation de chaque apprenant et se base sur les expériences, les connaissances et les questions particulières des participants pour nourrir le groupe de cas concrets et de retours d'expériences ciblées.

- Le format d'une journée réduit volontairement le périmètre aux fondamentaux de Terraform. Les sujets avancés, comme le multi-cloud, Terragrunt, Vault, Terratest ou Terraform Enterprise, sont présentés en ouverture ou en démonstration rapide.

Les supports de formation seront les suivants :

- Présentation théorique au format PowerPoint.
- Support d'exercices guidés au format PDF.
- Exemples de fichiers Terraform : main.tf, variables.tf, outputs.tf, providers.tf.
- Liens vers les documentations officielles de Terraform et des principaux Cloud Providers.

Ces supports seront fournis aux participants au cours de la formation au format PDF.

Méthodes d'évaluation des acquis

Avant la formation :

- Recueil éventuel des attentes et du contexte des participants en amont de la session.

En cours de formation :

- Points d'étapes réguliers par le formateur sur la compréhension des stagiaires.
- Vérification des résultats obtenus lors des exercices guidés.

Après la formation « à chaud » :

- Questionnaire d'auto-évaluation des compétences complété individuellement par chaque stagiaire après la formation.
- Questionnaire de satisfaction « à chaud » complété individuellement par chaque stagiaire en fin de formation.
- Le compte rendu formateur complété par le formateur.

Après la formation « à froid » :

- Questionnaire de satisfaction « à froid » complété individuellement par chaque stagiaire quelques semaines après la session de formation.

Un certificat de réalisation de fin de formation est remis au stagiaire lui permettant de faire valoir le suivi de la formation.