

Avant TerraForm

AWS CloudFormation

CloudFormation fournit par Amazon Web Service permet de créer et de gérer un ensemble de ressources qui sont liées, de les ordonner, les mettre en service et les actualiser en mode ordonné. Il permet d'avoir aussi une infrastructure as code avec des simples fichiers texte au format JSON ou YAML. Il fonctionne uniquement avec AWS mais le fonctionnement ressemble à terraform. Il permet de créer un modèle qui décrit toutes les ressources AWS que l'on veut (telles que des instances Amazon EC2 ou des instances de base de données Amazon RDS), et AWS CloudFormation s'occupe de leur mise en service et de leur configuration. AWS CloudFormation a des modèles d'exemples déjà créés qui peuvent être utilisés ou l'on peut en créer soit même.

Heat

Heat est un module de la partie orchestration de OpenStack. La mission du programme OpenStack Orchestration est de créer un service accessible pour gérer l'ensemble du cycle de vie des infrastructures et des applications dans le cloud OpenStack. Heat fournit une orchestration à base de modèle pour décrire une application cloud en exécutant appropriées OpenStack appels API pour générer l'exécution d'applications cloud. Un template Heat décrit l'infrastructure pour une application cloud dans des fichiers texte qui sont lisibles et modifiables par les humains, et peut être géré par des outils de contrôle de version. Le logiciel intègre d'autres composants de OpenStack. Les modèles permettent la création de la plupart des types de ressources OpenStack (tels que les instances, les instances flottantes, des volumes, des groupes de sécurité, les utilisateurs, etc.), ainsi que certaines fonctionnalités plus avancées telles que par exemple la haute disponibilité. Heat gère principalement l'infrastructure, mais les templates intègrent des outils de gestion de configuration logicielle tels que Puppet et Ansible.

TerraForm vs Autre Logicielle

Les outils comme CloudFormation, Heat, etc... permettent à une infrastructure d'être codifiés dans un fichier de configuration. Les fichiers de configuration permettent à l'infrastructure d'être élastiquement créés, modifiés et détruits. Terraform est inspiré par les problèmes qu'ils résolvent.

Terraform utilise de la même façon des fichiers de configuration pour détailler la configuration de l'infrastructure, mais il va plus loin par le diagnostic ainsi que la permission de fournisseurs multiples et des services combinés et composés. Par exemple, Terraform peut être utilisé pour orchestrer un AWS et un groupe OpenStack simultanément, en permettant des fournisseurs du 3ème parti (de 3ème fête) comme CloudFlare et DNSIMPLE être intégrés pour fournir des services de DNS et CDN. Ceci permet à Terraform de représenter et gérer l'infrastructure entière avec ses services de soutien, Au lieu de seulement le sous-ensemble

qui existe dans un fournisseur seul. Il fournit une syntaxe unifiée seule, au lieu d'exiger que des opérateurs utilisent des outils indépendants et non-interopérables pour chaque plateforme et service.

Terraform sépare également la phase de planification de la phase d'exécution, en utilisant le concept d'un plan d'exécution. En exécutant `terraform plan`, l'état actuel est actualisé et la configuration est consulté pour générer un plan d'action. Le plan comprend toutes les actions à entreprendre: quelles ressources seront créés, détruits ou modifiés. Il peut être contrôlé par les opérateurs pour assurer qu'elle est exactement ce qui est attendu. En utilisant `terraform graph`, le plan peut être visualisé pour montrer la commande dépendante. Une fois que le plan est capturé, la phase d'exécution peut être limitée aux seules actions du plan. D' autres outils combinent les phases de planification et d' exécution, ce qui signifie les opérateurs sont contraints de mentalement raisonner sur les effets d'un changement, qui devient rapidement insoluble dans les grandes infrastructures. Terraform permet aux opérateurs appliquent des changements avec confiance, car ils savent exactement ce qui se passera au préalable.

Conclusion