ATIR

L’infrastructure en tant que service : pratiques exemplaires

canarie.ca | @canarie_inc

Table des matières

Introduction

En épousant les pratiques décrites dans les pages qui suivent vous optimiserez les résultats obtenus avec l'infrastructure en tant que service (IaaS) de l'ATIR et les services d'infonuagique commerciaux. Voici les approches que nous préconisons.

Initialisation des instances

Les instances peuvent échouer sur l'ATIR comme sur n'importe quel serveur commercial. Il convient de considérer les instances comme un produit jetable qui ne contient ni données ni applications, ou dont la configuration est difficile à recréer. Advenant l'échec d'une instance, un des avantages de l'infonuagique est que le système en recréera une nouvelle en l'espace de quelques secondes ou minutes. Pour tirer parti de cette capacité, il faut cependant consacrer un peu de réflexion et d'efforts à la façon dont les instances sont initialisées.

Par « initialisation », on entend toutes les modifications que subit le serveur avant qu'il vous soit ou soit aux utilisateurs d'une réelle utilité. En prenant comme illustration la prestation de services Web, on devra, par exemple, commencer par installer et configurer le serveur Apache sur la nouvelle instance, puis vérifier l'application Django de Python à partir de la source et la configurer. Ces tâches sont habituellement effectuées à la main par le développeur ou l'administrateur du système qui prépare l'instance. Si l'instance plante, tout devra être repris manuellement. Il va donc dans votre intérêt d'automatiser l'initialisation des instances.

Il y a trois façons d'y parvenir.

Avec un script

On peut rédiger un script qui s'exécutera immédiatement au démarrage de l'instance. Ce script préparera cette dernière en vue de son utilisation. Le langage le plus utilisé pour cela est bash, mais des images acceptent aussi Python et Ruby.

Un script baptisé prep-app-server.sh pourrait ressembler à ceci.

```
#! /bin/bash
sudo apt-get install apache2 git
sudo sed -i “s/some config/my config/g” /etc/apache2/apache2.conf
git clone git://github.com/monentreprise/monprojet.git
sed -i “s/dev db/prod db/g” monprojet/settings.py
```

Pour lancer une instance par l'exécution du script immédiatement au démarrage, on recourrait à la commande suivante :

```
$ nova –-nom_clé mykey –-user-data prep-app-server.sh –-image “6f9bba21-5689-4e5c-af87-63c54a4ddbfe” –-flavor 2 boot nomdemoninstance
```

Le paramètre —user-data renvoie au fichier User Data renfermant les instructions qui initialisent l’instance.

La commande nova est une interface de ligne de commande OpenStack sur laquelle il convient de dire un mot.

Ce puissant outil permet de gérer l’environnement à distance.

```
Pour installer nova sur Centos :
$ sudo yum install epel-release
$ sudo yum install -y python-pip
$ sudo pip install --upgrade pip
$ sudo pip install python-novaclient
La commande nova est maintenant à votre disposition.
```

```
Pour installer nova sur Ubuntu :
$ sudo apt install python-pip
$ sudo pip install --upgrade pip
$ sudo pip install python-novaclient
La commande nova est maintenant à votre disposition.
```

Pour exécuter le script au lancement de l’instance à partir du Tableau de bord, utilisez l’onglet « Post-Création » de la fenêtre de démarrage. Il suffit de copier le script dans la zone texte « Script de Personnalisation » du dialogue au lancement de l’instance.

REMARQUE : Malheureusement, une telle automatisation est impossible avec les instances Windows.

Avec un instantané

Après avoir configuré l’instance manuellement, il est possible d’en prendre un instantané dont on se servira pour lancer de nouvelles instances identiques à l’originale. Cette méthode présente des avantages et des inconvénients. Le principal avantage est que l’instance calquée sur l’originale devrait être prête à servir dès le démarrage, sans autre intervention manuelle ni usage d’un script (lequel pourrait prolonger le téléchargement). Le principal inconvénient est que les décisions prises au moment où l’instance originale a été conçue feront toutes partie de la nouvelle instance.

Un autre aspect est que l’instantané ne peut inclure que les données du disque racine. On ne peut y intégrer les données des disques éphémères ni des volumes.

Pour vous, la solution se situe peut-être à mi-chemin entre l’utilisation d’un script et le regroupement des tâches. Une possibilité consisterait à installer tous les gros blocs de programmation dont vous avez besoin sur une instance, d’en prendre un instantané, puis de

configurer les blocs avec un script Post-Creation qui s'exécutera au lancement de l'instance à partir de l'instantané.

Avec des outils gérant la configuration des logiciels

Une nouvelle catégorie de logiciels connaît une popularité grandissante parce qu'ils gèrent la configuration d'autres logiciels sur les serveurs physiques et virtuels. Les outils de ce genre ne manquent pas. Mentionnons, entre autres, Chef, Puppet, Ansible, SaltStack et Jenkins, liste qui est loin d'être exhaustive. Les raisons pour recommander un outil ou un autre sont nombreuses, aussi en choisir un se résume parfois à une préférence personnelle. L'utilisation de ces outils déborde du cadre de ce document, mais l'adoption progressive de l'un d'entre eux contribuera beaucoup à automatiser l'infrastructure.

Sécurité

La sécurité a son importance dans n'importe quel environnement. La plupart des conseils en la matière s'appliquant à l'exploitation de vos serveurs physiques sont également valables pour les serveurs virtuels du nuage.

Pare-feu

Une exception notable concerne l'utilisation des groupes de sécurité dans l'ATIR. Les groupes de sécurité servent de pare-feu aux machines virtuelles (MV). Avant de lancer la première MV, on ouvrira l'onglet Groupes de sécurité du Tableau de bord et examinera les règles qui s'y trouvent. Ces règles servent à spécifier la nature et l'origine du trafic qui parviendra à votre MV. Modifier le CIDR, particulièrement, déterminera qui pourra se connecter à la MV, une puissante mesure de sécurité. Nous préconisons d'empêcher ceux qui utilisent le service d'accéder à la propriété intellectuelle d'origine en verrouillant celle-ci. Il est possible d'établir l'adresse IP publique de ceux qui utilisent votre service en effectuant une recherche Google avec « Quelle est mon adresse IP? »

Clés SSH

L'ATIR accepte les systèmes d'authentification à biclé (clé privée et clé publique), que nous recommandons vivement, car ces systèmes procèdent à une double authentification avant qu'on puisse se servir de la MV, si vous attribuez un NIP.
Lors de la création d'une biclé, la protection de la clé privée vous incombe entièrement. Une fois téléchargé sur la machine locale, le fichier de la clé privée est le seul en existence. Perdez-le et vous ne pourrez accéder aux MV lancées avec la biclé. Si le fichier est compromis, considérez que les MV lancées avec la biclé peuvent l'être également. Gardez vos clés privées en sécurité.

Voici une bonne liste pour rendre un environnement quelconque aussi sécuritaire que possible (adapté de Twenty Rules for Amazon Cloud Security)

1. Encryptez tout le trafic sur le réseau.
2. N'utilisez que des fichiers chiffrés pour les périphériques en mode bloc et les dispositifs locaux qui ne font pas partie du segment racine.
3. Encryptez au maximum tout ce que vous placez dans le stockage d'objets.
4. Ne laissez jamais les clés de chiffrement pénétrer dans le nuage, sauf pour procéder au décryptage et pas plus longtemps.
5. N'associez PAS de justificatifs d'identité aux images, à moins qu'il s'agisse d'une clé servant à décrypter la clé du système de fichiers.
6. Passez au système de fichiers chiffré avec une clé, au démarrage de l'instance.
7. N'autorisez jamais l'authentification par mot de passe pour accéder au système essentiel. Au grand jamais.
8. Ne donnez pas accès à sudo avec un mot de passe.
9. Concevez les systèmes de manière à ne pas devoir vous fier à une architecture d'images particulière pour faire fonctionner l'application.
10. Extrayez régulièrement une copie de sauvegarde complète du nuage et stockez-la ailleurs, dans un lieu sûr.
11. N'exploitez qu'un service par Instance.
12. N'ouvrez que le nombre minimum de ports voulus pour soutenir les services sur une instance.
13. Spécifiez l'adresse des sources au moment de configurer l'instance et n'autorisez un accès général qu'aux services globaux comme HTTP/HTTPS.
14. Séparez les données sensibles des autres données dans des bases et des groupes de sécurité différents quand vous hébergez une application aux données extrêmement délicates.
15. Automatisez les problématiques en sécurité. *(Vous en avez déjà connues à un moment ou un autre. Quand vous avez créé ce site FTP anonyme pour ouvrir les dossiers en lot que ce client vous envoie tous les soirs, par exemple.)*
16. Installez un système de détection des intrusions en mode hôte comme OSSEC.
17. Tirez parti des outils de renforcement du système comme Bastille, de Linux.
18. Si vous pensez que quelque chose a été compromis, sauvegardez le système de fichiers racine et supprimez l'instance. Pour procéderez à une analyse plus tard, sur un système non compromis.
19. Concevez les choses de façon à pouvoir appliquer un correctif de sécurité à une image, puis à relancer simplement les Instances.
20. Par-dessus tout, créez des applications Web sécuritaires.

Application de correctifs

Tôt ou tard, la sécurité des logiciels devra être mise à niveau. C'est une nécessité. La manière de diffuser les correctifs et le moment de le faire dépendent de la tolérance de l'application aux changements. Habituellement, la plupart des correctifs sont compatibles avec les versions antérieures, de sorte qu'on peut les appliquer dès qu'ils deviennent disponibles, si on le désire. Cependant, il arrive qu'une application dépende d'un détail infime, dans un logiciel que modifiera une rustine de sécurité scindant l'application. Puisque l'application que vous avez créée n'a pas de secrets pour vous, vous êtes le seul à pouvoir dire comment et quand les correctifs devraient être appliqués. Si vous redoutez l'impact d'un correctif, prenez un instantané de l'instance, appliquez le correctif et assurez-vous que tout va bien. Si c'est le cas, vous n'aurez qu'à supprimer l'instantané. Si des problèmes surgissent, faites marche arrière.

Une approche consiste à faire confiance au fournisseur du système d'exploitation et à autoriser les correctifs automatiques. Voici comment procéder aux mises à jour de sécurité automatiques sur les différentes plateformes d'exploitation utilisées par l'ATIR.

- Ubuntu
- CentOS
- Windows

Une approche plus conservatrice serait de ne recevoir que les avis signalant l'existence d'un correctif. Ainsi, vous pourrez planifier les mises à jour et éventuellement les tester dans un environnement de développement avant leur application à l'environnement d'essai.

- Ubuntu
- CentOS
- Windows

Toutes les distributions Linux dans l'ATIR signaleront l'existence de correctifs à l'ouverture de la séance par défaut. L'application automatique de correctifs est désactivée dans Windows.

Noyaux

« Vous n'avez pas à trop vous inquiéter au sujet de problèmes de sécurité dans le noyau. Peu de problèmes surviennent à ce niveau. Leur compatibilité ou leur stabilité avec la plateforme d'exploitation revêtent beaucoup plus d'importance.

Si le fournisseur propose des noyaux plus récents et que vous prévoyiez de la maintenance de toute manière, vous débarrasser d'un noyau avec la commande que vous avez établie est une bonne idée. Si vous utilisez les noyaux Amazon, je ne m'en soucierai pas. Évitez d'y toucher. À moins d'un besoin particulier, je ne me préoccuperais pas d'utiliser un noyau spécial dans votre AMI. »

Adapté de : Do I need to manually administer EC2 kernel updates?

Application

Le système d'exploitation est aussi sûr que les applications qu'il héberge (celles que vous utilisez ou que vous développez). Bien que la sécurité des applications déborde du cadre de ce document, nous vous suggérons de visiter le site de l'Open Web Application Security Project, un excellent point de départ pour le développement d'applications Web sécuritaires.

Volumes

Stockez vos données sur un volume!

Les données, les applications ou les configurations qui ne peuvent être recréées facilement seront stockées sur un volume. Par défaut, l'instance démarre de son disque racine **(/dev/vda)** et d'un disque éphémère **(/dev/vdb)**, mais les données n'y sont stockées que de façon transitoire. Elles n'y demeurent pas. Quand on met fin à l'instance, les données sur ces deux disques sont irrémédiablement perdues. En revanche, celles stockées dans un volume survivront à l'instance et le volume pourra être attaché à une nouvelle instance, ce qui vous permettra de poursuivre votre travail.

Quand vous annexez un volume à une instance avec le tableau de bord ou la ligne de commande, sachez que le système ne tient pas compte du dispositif spécifié. Le volume sera annexé au premier dispositif disponible dans **/dev/vd[a-z] (le plus souvent 'c')**. Après dissociation, ce dispositif ne sera pas réutilisé avant la réinitialisation de l'instance *(en d'autres termes, le volume sera annexé au dispositif suivant, jusqu'à ce qu'on arrive à la fin de la chaîne)*. Quand l'instance est réinitialisée à l'interne *(quand l'utilisateur la relance avec la commande correspondante à partir d'un terminal situé dans l'instance)*, les volumes seront annexés à un nouveau dispositif, conformément à la règle précitée, à moins que les dispositifs se trouvent déjà dans cet ordre.

/etc/fstab

Faites très attention quand vous utilisez la commande fstab pour gérer le montage automatique des volumes, car le nom du dispositif attribué au volume peut changer. Avec fstab, utilisez plutôt des étiquettes ou des UUID.

Ajouter nobootwait aux options a aussi son utilité, car ainsi, l'instance ne gèlera pas si le volume correspondant s'est détaché pour une raison quelconque. La commande errors=remount empêchera aussi le disque de se corrompre, advenant une panne du réseau ou du serveur.

Voici les étapes à suivre à la création pour ajouter sans risque un volume à fstab. Sauter la première étape pour les volumes existants.

1. Créer le volume, l'annexer à l'instance, créer le système de fichiers souhaité, puis procéder d'une des deux façons que voici.
2. (Facultatif) Donner une étiquette au système de fichiers
    a. Cette étape dépend de la nature du système; voir la liste des utilitaires à : https://wiki.archlinux.org/index.php/Persistent_block_device_naming - By-label.
    b. Cette étape peut aussi être effectuée à la création du système de fichiers en spécifiant L <label> avec mkfs (voir la page principale du système de fichiers)

```
$ mkfs <fstype>
```

    c. Pour retrouver une étiquette oubliée, utiliser :

```
ls -l /dev/disk/by-label or blkid /dev/XXX
```

3. (Facultatif) Trouver l'UUID

```
ls -l /dev/disk/by-uuid or blkid /dev/XXX
```

4. Au moment d'ajouter une entrée à fstab, indiquer le système de fichiers (première colonne) portant l'étiquette ou l'UUID, selon ce qu'on a choisi.
    a. LABEL=<étiquette>
    b. UUID=<UUID>

Exemples :

```
UUID=c4d66b8a-2984-438b-8004-394613d50c30  /mnt  auto  nobootwait  0  0
LABEL=MON_VOLUME  /some/dir  ext3  nobootwait  0  0
```

Une autre solution, si nobootwait n'a pas fonctionné et que l'instance fige parce qu'elle n'arrive pas à accéder au volume, consiste à rédiger un script de démarrage et de fermeture qui annexera le volume à l'instance et l'en détachera. Cette solution fera aussi en sorte que les volumes sont détachés correctement à la réinitialisation de l'instance.

Démarrage : ajouter ce qui suit à /etc/rc.local, au-dessus de la ligne exit 0.

/root/attach-volumes.sh

Ensuite, créez /root/attach-volumes.sh:

```
#!/bin/bash
source /path/to/your/rc/file

nova volume-attach “id serveur” “id volume” “/dev/vdc”
sleep 4
mount -L <LABEL /your/mount/point
```

Reprendre ces trois lignes en fonction du nombre de volume que vous voulez annexer.

Pour le script de fermeture, créez le fichier /etc/init.d/detachvolumes

```
#! /bin/bash
### BEGIN INIT INFO
# Provides:          detachvolumes
# Required-Start:
# Required-Stop:
# Default-Start:
# Default-Stop:      0 6
# Short-Description: Détacher les volumes externes à la fermeture et réinitialiser
# Description:
### END INIT INFO

PATH=/sbin:/usr/sbin:/bin:/usr/bin
. /lib/init/vars.sh

. /lib/lsb/init-functions

umask 022

do_stop () {
        # umount fs
        umount /your/mount/point
        # then detach external volumes
        source /path/to/your/rc/file
        sleep 1
        nova volume-detach “id serveur” “id volume”
        sleep 1
}

case "$1" in
  start)
        # No-op
        ;;
  restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
  stop)
        do_stop
```

```
        ;;
 *)
        echo "Usage: $0 start|stop" >&2
        exit 3
        ;;
esac
```

Puis autorisez l'exécution de ce script à la fermeture :

```
$ update-rc.d detachvolumes stop 22 0 6 .
```

Sauvegardes

Sauvegardez régulièrement vos volumes!

Les volumes peuvent se détériorer et ils le feront. Effectuez-en régulièrement une copie de sauvegarde dans une autre partie de l'ATIR, dans d'autres volumes du nuage ou dans un volume local avec rsync. Si vous utilisez une base de données, déterminez comment effectuer des copies de sauvegarde régulièrement avec le logiciel correspondant.

Architecture

Établir l'architecture qui conviendra le mieux à votre application déborde du cadre de ce document. Néanmoins, voici quelques ressources qui vous aideront dans cette tâche.

- AWS Architecture Center
- Cloud Application Architecture
- Scaling Experts