

03 - IAM, sécurité, stockage et observabilité

Objectif : comprendre les mécanismes essentiels de sécurité et d'administration Scaleway, savoir configurer les accès, protéger les secrets, gérer les quotas et superviser les ressources.

Périmètre : IAM, API keys, Organizations, Projects, quotas, Object Storage, KMS, Secret Manager, Cockpit, Audit Trail et alertes de facturation.

1. Organisation, Projects et ressources

1.1 Organization

L'Organization est le périmètre principal qui regroupe les membres, les applications IAM, les politiques, les Projects et les ressources associées au compte.

- **Owner** : propriétaire de l'Organization, avec des responsabilités d'administration spécifiques.
- **Member** : membre disposant de permissions selon sa configuration IAM.
- **User** : identité humaine au sein de l'Organization.
- **Application** : identité non humaine destinée notamment aux automatisations et aux API keys.

1.2 Projects

Un Project sert à organiser les ressources et à séparer les environnements, les applications ou les équipes. Un Project peut regrouper des ressources situées dans plusieurs régions et Availability Zones.

Exemple d'organisation : deux environnements, chacun avec deux équipes produit, peuvent être séparés en quatre Projects :

- Développement — équipe A
- Développement — équipe B
- Production — équipe A
- Production — équipe B

Calcul : 2 environnements × 2 équipes = 4 Projects.

1.3 Ajouter une ressource depuis la console

1. Sélectionner le Project dans lequel créer la ressource.
2. Choisir la région ou l'Availability Zone si le service le demande.
3. Configurer les paramètres de la ressource.
4. Vérifier les paramètres et lancer la création.

Attention : les possibilités de déplacement d'une ressource vers un autre Project dépendent du service. Ne pas supposer que toutes les ressources peuvent être transférées directement : vérifier la procédure officielle du produit concerné.

2. IAM : identités et permissions

2.1 Les concepts fondamentaux

Concept	Rôle
User	Identité humaine appartenant à une Organization.
Application	Identité non humaine utilisée notamment pour l'automatisation.
API key	Identifiant et secret utilisés pour authentifier les requêtes API.
Policy	Définit les permissions accordées à un principal dans un périmètre donné.
Group	Regroupe des utilisateurs et des applications pour partager des policies.
Principal	Identité ou groupe auquel une policy est associée.

Une policy définit notamment :

- Le principal concerné.
- Le périmètre d'application (scope).
- Les permissions ou ensembles de permissions accordés.

Le scope peut notamment être défini au niveau de l'Organization ou d'un Project, selon la configuration souhaitée.

2.2 Groupes et retrait d'un membre

Les groupes permettent d'appliquer une même policy à plusieurs identités.

Lorsqu'un utilisateur est retiré d'un groupe, il perd les permissions héritées de ce groupe. Il peut toutefois conserver les permissions qui lui sont accordées directement ou par d'autres groupes.

Le retrait d'un groupe ne signifie donc pas nécessairement que l'utilisateur perd tout accès à l'Organization.

2.3 Les API keys

Une API key Scaleway est composée de deux éléments :

- **Access key** : identifiant de la clé, comparable à un nom d'utilisateur.
- **Secret key** : valeur confidentielle utilisée pour authentifier les requêtes. Elle doit être protégée comme un mot de passe.

Une API key hérite des permissions de son bearer, c'est-à-dire l'utilisateur ou l'application auquel elle est associée.

Une API key est rattachée à une seule Organization. Si plusieurs Organizations doivent être utilisées, une clé distincte est nécessaire pour chacune.

2.4 Utilisateur ou application comme bearer ?

Pour une automatisation ou un service de production utilisant une clé sur le long terme, privilégier une **application IAM**.

- Une clé liée à un utilisateur dépend de l'identité humaine correspondante.
- Une clé liée à une application n'est pas liée à un membre humain particulier.
- Les permissions de l'application doivent être explicitement configurées via les policies IAM.

Les applications ne sont pas des comptes humains et ne sont pas liées à un utilisateur particulier.

2.5 Exemple : configuration Terraform avec une API key

Exemple de configuration du provider Scaleway. La clé doit être fournie par un mécanisme sécurisé, et non inscrite en clair dans un dépôt Git.

```
terraform {  
  required_providers {  
    scaleway = {
```

```
        source = "scaleway/scaleway"
    }
}

provider "scaleway" {
    # Fournir les identifiants par variables d'environnement
    # ou par un mécanisme sécurisé adapté.
    region = "fr-par"
    zone   = "fr-par-1"
}
```

Exemple d'utilisation de variables d'environnement dans un shell :

```
export SCW_ACCESS_KEY="VOTRE_ACCESS_KEY"
export SCW_SECRET_KEY="VOTRE_SECRET_KEY"
export SCW_DEFAULT_PROJECT_ID="VOTRE_PROJECT_ID"

terraform init
terraform plan
```

Sécurité : ne pas committer les secrets, limiter les permissions au strict nécessaire et protéger le fichier d'état Terraform, qui peut contenir des données sensibles.

2.6 IAM et facturation

IAM est un service gratuit. Cela ne signifie pas que les ressources créées ou utilisées grâce à IAM sont gratuites.

IAM fonctionne avec les produits Scaleway compatibles. Pour vérifier la compatibilité exacte d'un service ou d'une action, consulter la documentation IAM du produit concerné.

3. Quotas et limites de l'Organization

3.1 Comprendre les quotas

Les quotas limitent le nombre ou la capacité des ressources utilisables. Ils sont associés à l'Organization et peuvent varier selon le produit, le type de ressource, la région et le niveau du compte.

Les quotas ne sont généralement pas des limites personnelles appliquées séparément à chaque utilisateur. Plusieurs utilisateurs d'une même Organization peuvent donc consommer le quota partagé du produit concerné.

3.2 Augmenter les quotas

Les deux actions principales permettant d'augmenter le niveau de compte sont :

1. Valider le moyen de paiement.
2. Vérifier son identité.

Si les quotas restent insuffisants après ces vérifications, contacter le support Scaleway pour demander une augmentation supplémentaire.

La console permet de consulter les quotas, leur utilisation et les limites applicables. Le bouton d'augmentation éventuellement proposé dépend du produit et du quota concerné.

Important : les limites par défaut peuvent évoluer. Vérifier la documentation officielle avant de mémoriser une valeur numérique telle que le nombre maximal de Projects.

4. Control plane et Data plane

4.1 Control plane

Le Control plane correspond à la couche de gestion et d'orchestration des ressources. Elle permet notamment de créer, configurer et administrer les ressources au moyen de la console ou des API.

Exemples :

- Créer une Instance.
- Configurer une ressource depuis la console.
- Appeler une API pour gérer une ressource.

4.2 Data plane

Le Data plane correspond à la couche opérationnelle où les ressources sont utilisées et exécutent leur travail.

Exemples :

- Se connecter à une Instance Linux via SSH.
- Exécuter une application sur une Instance.

- Traiter le trafic de l'application.

Mémo : Control plane = gérer et orchestrer ; Data plane = utiliser et exécuter.

5. Secret Manager et secrets éphémères

5.1 À quoi sert Secret Manager ?

Secret Manager sert à stocker et gérer des données sensibles telles que des mots de passe, des tokens, des clés API et des identifiants d'applications.

Un secret peut posséder plusieurs versions. Les versions peuvent être activées ou désactivées pour contrôler leur accessibilité.

La taille maximale documentée d'un secret est de **64 kB**.

5.2 Accorder un accès à une application

Pour autoriser une application à lire un secret :

1. Identifier l'application IAM qui doit accéder au secret.
2. Créer ou configurer une policy IAM adaptée.
3. Accorder uniquement les permissions nécessaires à l'application et au périmètre concerné.
4. Tester l'accès avec l'application.

Ne jamais partager des secrets en clair avec tous les membres d'une Organization simplement pour faciliter l'accès.

5.3 Politiques éphémères

Une politique éphémère définit la durée de vie ou les conditions d'expiration des versions d'un secret.

Politique	Fonctionnement
Single access	La version du secret expire après un accès unique.
Time to Live (TTL)	La version reste valide et accessible pendant une durée définie.

La durée maximale documentée pour une politique TTL est **d'un an**.

Point important : la politique éphémère doit être configurée lors de la création du secret. Elle ne peut pas être ajoutée après coup à un secret existant et ne peut pas être retirée une fois appliquée.

5.4 Exemple de stockage sécurisé

Exemple de variables d'environnement utilisées par une application. Les valeurs sont fictives et ne doivent pas être remplacées par de vrais secrets dans un fichier versionné.

```
DATABASE_HOST="db.example.internal"
DATABASE_NAME="application"
DATABASE_USER="app_user"
DATABASE_PASSWORD="A_INJECTER_DEPUIS_SECRET_MANAGER"
```

En production, récupérer le mot de passe depuis Secret Manager ou par un mécanisme d'injection sécurisé adapté à l'environnement d'exécution.

6. KMS et chiffrement

6.1 Rôle de KMS

KMS (Key Management Service) sert à créer et gérer des clés cryptographiques. Il ne remplace pas Secret Manager pour le stockage de mots de passe, de tokens ou de secrets applicatifs.

- **KMS** : gestion des clés cryptographiques.
- **Secret Manager** : stockage et gestion des secrets.

6.2 Chiffrer de grands volumes de données

Pour chiffrer de grands volumes de données, on utilise généralement le chiffrement par enveloppe :

1. Une clé de données (DEK, Data Encryption Key) chiffre les données.
2. Une clé gérée ou protégée par KMS protège la DEK.
3. La DEK protégée est stockée avec les métadonnées nécessaires au déchiffrement.
4. Lors du déchiffrement, la DEK est récupérée de façon sécurisée puis utilisée pour déchiffrer les données.

Cette approche évite de demander à KMS de chiffrer directement chaque bloc d'un grand jeu de données.

6.3 Perte d'une clé

Si la clé nécessaire au déchiffrement est perdue et qu'aucune sauvegarde ou procédure de récupération n'existe, les données peuvent devenir irrécupérables.

Il faut prévoir une stratégie de sauvegarde, de contrôle d'accès et de récupération des clés avant de mettre en production un système de chiffrement.

7. Object Storage et chiffrement SSE

7.1 Quand utiliser Object Storage ?

Object Storage stocke des objets dans des buckets. Il est adapté notamment aux fichiers, images, archives, sauvegardes, journaux et données applicatives non structurées.

Il peut être utilisé par des applications et des solutions de calcul pour lire ou écrire des objets. Il ne remplace pas une base de données relationnelle lorsqu'il faut exécuter des requêtes SQL sur des données structurées.

7.2 Les méthodes SSE

Méthode	Principe
SSE-S3	Chiffrement côté serveur avec des clés gérées par le service de stockage.
SSE-C	Chiffrement côté serveur avec une clé fournie par le client.
SSE-KMS	Terme générique désignant un chiffrement reposant sur un service de gestion de clés ; vérifier les méthodes effectivement prises en charge par le service.

Pour SSE-C, le serveur effectue le chiffrement, mais le client fournit la clé. La clé doit être protégée et conservée de manière fiable, car sa perte peut empêcher le déchiffrement.

7.3 Sécuriser l'accès aux buckets

La sécurité d'un bucket peut impliquer plusieurs mécanismes :

- Les permissions IAM.

- Les bucket policies lorsqu'elles sont adaptées au besoin.
- Les paramètres de visibilité et d'accès du bucket.
- La gestion des clés et des identifiants utilisés par les applications.

Les bucket policies ne doivent pas être considérées comme l'unique mécanisme de sécurité. La configuration exacte dépend du type d'accès recherché et des fonctionnalités du service.

En cas de refus d'accès, examiner l'ensemble des règles applicables, notamment les permissions IAM, les bucket policies et les paramètres de visibilité. Ne pas supposer qu'une autorisation présente dans une policy suffit à neutraliser un refus provenant d'un autre mécanisme.

8. Cockpit : métriques, logs et supervision

8.1 Rôle de Cockpit

Scaleway Cockpit fournit des fonctions d'observabilité permettant de collecter, stocker et analyser des données de supervision. Il peut recevoir des métriques et des logs provenant de ressources Scaleway et de systèmes compatibles externes.

8.2 Métriques, logs et traces

Type	Utilité	Exemple
Metrics	Mesures numériques dans le temps.	CPU, mémoire, réseau, latence.
Logs	Événements et messages détaillés.	Erreur d'application, échec de démarrage, message système.
Traces	Parcours d'une requête à travers les services.	Identifier l'étape qui ralentit une requête distribuée.

Les logs sont souvent les plus utiles pour rechercher les erreurs et les événements intervenus pendant le cycle de vie d'une ressource.

8.3 Surveiller une Instance ou un serveur Elastic Metal

Les métriques disponibles peuvent inclure l'utilisation du CPU, de la mémoire, du disque et du réseau. Leur disponibilité exacte dépend du service, de la configuration et des agents utilisés.

Pour les Instances Scaleway, Cockpit est l'outil intégré à privilégier pour consulter les métriques de supervision disponibles.

8.4 Envoyer des métriques avec Prometheus

Prometheus peut envoyer des métriques vers Cockpit grâce à l'interface Prometheus Remote Write.

Exemple de configuration indicative :

```
global:
  scrape_interval: 60s

remote_write:
  - url: https://metrics.cockpit.fr-par.scw.cloud/api/v1/push
    headers:
      X-Token: <VOTRE_TOKEN_COCKPIT>
```

Remplacer l'URL par le point de terminaison correspondant à la région Cockpit utilisée et injecter le token de manière sécurisée. Cet exemple illustre le principe ; il doit être adapté à la configuration réelle.

8.5 Périmètre des données Cockpit

Les données Cockpit sont associées à un périmètre de supervision lié au Project dans le contexte présenté par le cours. Il faut donc vérifier le Project sélectionné lorsqu'une métrique semble absente.

Pour diagnostiquer une métrique manquante, contrôler :

1. Le Project et la région sélectionnés.
2. La configuration de l'agent ou de Prometheus.
3. Le point de terminaison Remote Write.
4. Le token et les permissions nécessaires.
5. La présence de données dans la période consultée.

9. Alertes et facturation

9.1 Alert Manager

Le service d'alertes doit être activé et configuré. Ne pas supposer que l'activation entraîne automatiquement l'activation de toutes les alertes préconfigurées dans toutes les régions.

Après activation, vérifier les alertes disponibles, leur état, leur périmètre et les conditions de déclenchement.

9.2 Alertes de facturation

Pour être averti lorsque la consommation et les coûts augmentent, configurer les alertes de facturation dans la section Billing de la console.

Les alertes CPU dans Cockpit ou Grafana servent à surveiller les performances techniques. Elles ne remplacent pas les alertes de facturation.

9.3 Audit Trail

Audit Trail conserve des événements permettant de savoir qui a effectué une action, quand elle a eu lieu et si la requête a réussi ou échoué.

Les événements sont stockés dans la région où l'activité s'est produite. Les événements liés aux produits globaux, comme IAM, sont stockés dans la région de Paris selon la documentation du service.

Pour rechercher un événement, filtrer notamment par région, période, Project, produit, méthode API et statut de la requête.

Audit Trail sert à l'audit et au dépannage. Il ne remplace pas les logs applicatifs ou les métriques de performance.

10. Comprendre les accès et les refus

10.1 Principe du moindre privilège

Accorder à chaque utilisateur ou application uniquement les permissions nécessaires à ses tâches.

Exemple :

- Une application qui doit lire un secret ne devrait pas recevoir automatiquement des droits d'administration sur toute l'Organization.
- Une équipe de développement peut être limitée à ses Projects.

- Une clé API de production doit utiliser une application dédiée avec des permissions limitées.

10.2 Méthode de dépannage

1. Identifier le principal qui effectue la requête.
 2. Identifier le bearer de l'API key et les permissions qui lui sont attribuées.
 3. Vérifier les politiques attachées directement et via les groupes.
 4. Contrôler le scope de chaque policy.
 5. Vérifier les règles spécifiques du service, comme les bucket policies.
 6. Consulter Audit Trail et les logs pour obtenir des indices sur le refus.
-

11. Pièges du QCM à retenir

- **API key** : elle hérite des permissions de son bearer et reste limitée à une Organization.
 - **Clé API de longue durée** : préférer une application IAM à un utilisateur humain.
 - **Secret key** : c'est la partie la plus sensible de l'API key.
 - **IAM** : le service est gratuit ; les ressources utilisées peuvent être payantes.
 - **Quota** : les niveaux de compte dépendent notamment de la validation du moyen de paiement et de l'identité.
 - **Control plane** : gestion et orchestration des ressources.
 - **Data plane** : utilisation opérationnelle des ressources, par exemple une connexion SSH à une Instance.
 - **KMS** : gestion des clés cryptographiques, pas stockage des secrets applicatifs.
 - **DEK** : clé de données utilisée pour chiffrer les données dans le chiffrement par enveloppe.
 - **Secret Manager** : une politique éphémère est configurée à la création du secret ; le TTL peut aller jusqu'à un an.
 - **Object Storage** : stockage d'objets et de fichiers ; il peut être consommé par des applications et des solutions de calcul.
 - **SSE-C** : chiffrement côté serveur avec une clé fournie par le client.
 - **Cockpit** : métriques et logs ; Prometheus peut envoyer des métriques via Remote Write.
 - **Logs** : messages d'événements et erreurs ; métriques = mesures chiffrées ; traces = parcours des requêtes.
 - **Billing Alerts** : alertes de facturation, distinctes des alertes CPU et mémoire.
 - **Audit Trail** : événements d'audit de l'Organization, stockés selon la région de l'activité.
 - **Déplacement de ressources** : les possibilités varient selon le service ; vérifier la procédure du produit avant de répondre à une question générale.
-

12. Liens officiels utiles

- [Documentation IAM](#)
- [Créer des API keys](#)
- [Quotas de l'Organization](#)
- [Secret Manager — démarrage rapide](#)
- [Limites de Secret Manager](#)
- [Envoyer des métriques et logs vers Cockpit](#)
- [FAQ Audit Trail](#)

Conclusion

Pour sécuriser une infrastructure Scaleway, il faut combiner des identités bien définies, des politiques IAM limitées au besoin, des API keys protégées, une gestion appropriée des secrets et des clés, ainsi qu'une supervision adaptée.

Les valeurs numériques et les fonctionnalités propres à un produit peuvent évoluer. En cas de doute, utiliser la documentation officielle du service plutôt que de généraliser une règle observée sur un autre produit.

Revision #1

Created 2026-10-09 12:52:07 UTC by Nico là

Updated 2026-10-09 12:53:25 UTC by Nico là