

01 - Foundations

?? 1. Les grandes familles

Besoin	Produit à retenir
VM classique	Instances
Serveur physique dédié	Elastic Metal
GPU / IA / rendu / encodage	GPU Instances
ARM / Ampere	CAX
Mac Apple Silicon	Apple Silicon / Mac mini
Kubernetes managé	Kapsule
Conteneur serverless	Serverless Containers
Fonction serverless	Serverless Functions
Fichiers / objets	Object Storage
Disque attaché à une VM	Block Storage
Archivage froid	Glacier
Base SQL	Managed Databases
Cache / sessions	Redis
Analytics / gros volumes	ClickHouse
Répartition du trafic	Load Balancer
IP publique portable	Flexible IP
Monitoring / observabilité	Cockpit
Identités / permissions	IAM
Messages pub/sub	Topics & Events
Messages point-to-point	Queues
Emails transactionnels	Transactional Email
IA générative par API	Generative APIs
Infrastructure as Code	Terraform

?? 2. Instances

Development Instances

- Développement / tests
- **vCPU partagés**

General Purpose

- Usage général
- C'est notamment là qu'on trouve **CAX**, avec processeurs **Ampere ARM**.

GPU Instances

- **IA**
- Data processing
- Rendering
- Video encoding

Elastic Metal

- **Bare Metal**
- Serveur physique, pas une VM.

Apple Silicon

- Mac mini Apple Silicon
 - Connexion notamment **SSH + VNC**
 - Attention au **minimum de facturation de 24 h**.
-

? 3. Storage

Object Storage

Imagine :

Bucket → Objects

Un bucket peut contenir énormément d'objets, pas seulement 100.

Les objets sont **privés par défaut**.

Versioning

Si tu modifies un objet avec le versioning activé :

```
photo.jpg
  ↓ modification
photo.jpg version 1
photo.jpg version 2
photo.jpg version 3
```

→ Les anciennes versions sont conservées.

→ Elles consomment donc du stockage → **coût supplémentaire**.

Site web dans Object Storage

Pour personnaliser le domaine :

CNAME

Block Storage

Pense :

“ **Block Storage = disque** ”

Tu peux :

attach → attacher le volume à une Instance

detach → le détacher

Très différent d'Object Storage.

Glacier

Pense :

“ Glacier = archivage / stockage froid

Et surtout :

❑ **Pas de Multi-AZ redundancy** comme le stockage Standard Multi-AZ.

?? 4. Databases

PostgreSQL / MySQL

- données structurées
- applications classiques

Redis

- **mémoire / vitesse**
- cache
- sessions
- données temporaires
- API

Question typique :

“ "Best database for storing sessions?"

❑ **Redis**

ClickHouse

- Analytics
- Très gros volumes de données
- Data science / Business Intelligence

Et :

“ **Data Warehouse = centralized repository for large sets of structured data**

?? 5. Kubernetes

Pour l'instant, retiens simplement :

Kapsule = Kubernetes managé par Scaleway

Le cluster est intégré à l'écosystème Scaleway.

Et surtout :

“ **Terraform peut gérer un cluster Kapsule.**

On va approfondir ça juste après avec la certification **Scaleway Associate Kubernetes**. ☐

? 6. Serverless

Serverless Functions

- Tu déploies une fonction
- Scaleway gère l'infrastructure
- Ça scale automatiquement.

Quand il n'y a plus d'activité :

requêtes

↓

instances

↓

0

Puis une nouvelle requête arrive.

Le temps nécessaire pour redémarrer :

☐ **Cold start**

Point important

Une instance de Serverless Function ne traite pas plusieurs requêtes simultanément.

→ Scaleway peut créer plusieurs instances pour gérer la charge.

Serverless Containers

→ Tu fournis un **container**

→ Scaleway gère l'infrastructure

→ Possibilité d'utiliser des **encrypted environment variables** pour les informations sensibles.

? 7. Network

Load Balancer

→ Distribue les requêtes entre plusieurs ressources.

Il peut être :

☐ **Public**

☐ **Private**

Flexible IP

C'est une IP :

public + portable + persistante

Tu peux :

Flexible IP



Instance A



detach



Instance B

Elle est indépendante de la ressource.

⚠ Elle est liée à une **Availability Zone**.

? 8. IAM

IAM =

“ Identity and Access Management

Il sert à gérer :

qui → peut faire quoi → sur quelles ressources

Organization

L'Organization regroupe les utilisateurs et ressources.

Owner

→ propriétaire de l'Organization.

Members / Users

→ utilisateurs auxquels on attribue des permissions.

Policies

→ définissent les permissions.

API Keys

Point à retenir :

“ **API keys inherit the permissions of their bearer.** ”

Donc une API key n'est pas magiquement plus puissante que l'utilisateur auquel elle appartient.

MFA

→ authentification multi-facteurs.

? 9. Cockpit

Cockpit =

“ **Observability** ”

Il permet de centraliser :

- métriques
- logs
- dashboards
- alertes

Et surtout :

☐ **pré-built dashboards** pour les produits compatibles.

Cockpit n'est pas limité aux produits Scaleway : il peut aussi recevoir des données externes.

? Facturation Cockpit

À retenir pour le QCM :

☐ consultation d'un dashboard → pas facturée

☐ création d'une alerte → pas facturée

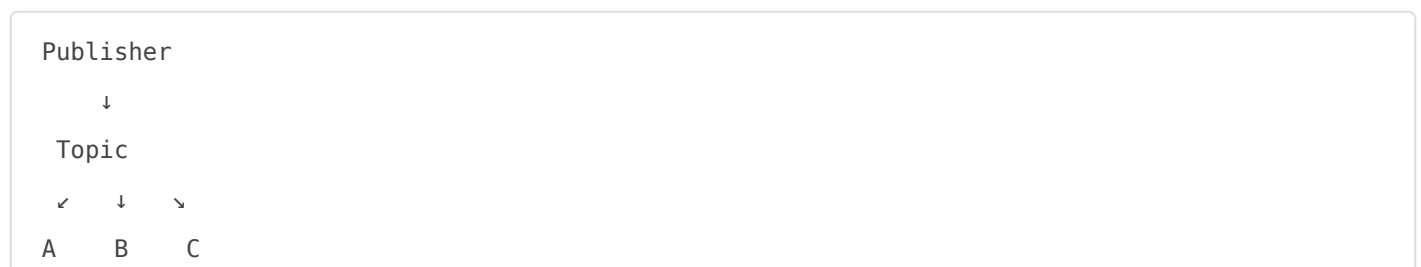
⚠ rétention au-delà de la limite → facturée

? 10. Messaging

Deux concepts à ne surtout pas mélanger :

Topics & Events

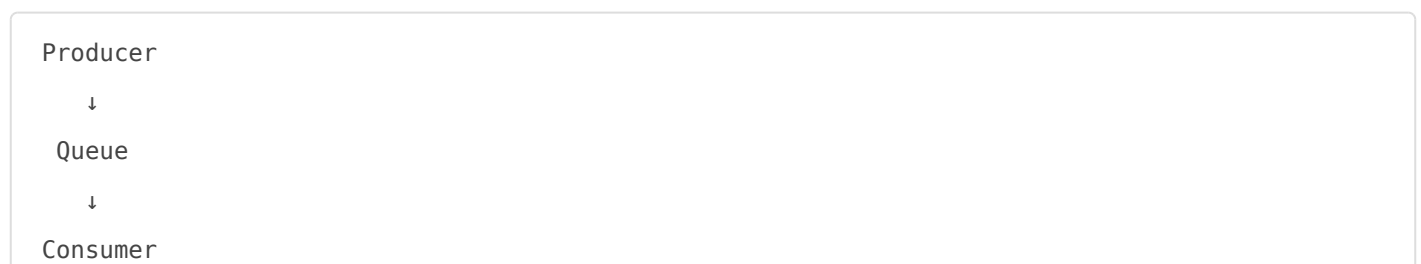
☐☐ **Publish / Subscribe**



Plusieurs consommateurs peuvent recevoir le message.

Queues

☐☐ **Point-to-point**



→ Un message est destiné à être traité par un consommateur.

☐ Mémo :

Topics = broadcast

Queues = file d'attente

? 11. Transactional Email

Workflow :

1. Create account
- ↓
2. SPF / DKIM / DMARC
- ↓
3. Integrate SMTP / API
- ↓
4. Send
- ↓
5. Monitor

Donc :

Create → Authenticate → Integrate → Send → Monitor

? 12. AI

GPU Instances

→ GPU à la demande.

AI Supercomputers

→ infrastructures GPU massives pour l'IA.

Generative APIs

→ accès aux modèles génératifs via API.

Facturation :

“ Pay-as-you-go / tokens

Donc :

plus de tokens → plus de consommation

?? 13. Terraform

Terraform = **Infrastructure as Code**

Au lieu de :

Console

→ clic

→ clic

→ clic

tu décris :

Terraform

→ configuration

→ plan

→ apply

Bonne pratique importante

Évite de mélanger :

création manuelle dans Console + gestion Terraform

Si une ressource existe déjà :

→ **import dans Terraform state**

→ puis gestion par Terraform.

? 14. Region vs Availability Zone

Très important pour les QCM :

Region

→ zone géographique.

Exemple :

Paris

Availability Zone

→ zone d'infrastructure isolée au sein d'une région.

Et attention à la phrase :

“ "Each Region is composed of two Availability Zones."

❑ **False**

Il ne faut surtout pas mémoriser "1 région = 2 AZ".

? 15. Billing

À retenir :

Facturation mensuelle

La consommation des services est généralement regroupée dans une **facture mensuelle**.

Mais certains éléments peuvent avoir des modalités particulières.

Exemple :

Domain names et certains **Elastic Metal setup fees** peuvent être facturés lors de la commande.

? 16. Les associations à connaître PAR CŒUR

Si tu ne devais retenir que ça :

Kubernetes	→ Kapsule
Sessions	→ Redis
Analytics	→ ClickHouse
Structured data	→ Managed Database
Objects	→ Bucket
Disk	→ Block Storage
Archive	→ Glacier
GPU / AI	→ GPU Instances
ARM / Ampere	→ CAX
Bare Metal	→ Elastic Metal
Monitoring	→ Cockpit
Permissions	→ IAM
Public IP	→ Flexible IP
Traffic	→ Load Balancer
Pub/Sub	→ Topics & Events
Point-to-point	→ Queues
IaC	→ Terraform
Email	→ Transactional Email
LLM API	→ Generative APIs

Et surtout :





Bucket = Object Storage
Volume = Block Storage
Database = données structurées
Redis = vitesse / sessions
Kapsule = Kubernetes

Revision #1

Created 2026-08-28 06:23:27 UTC by Nico là

Updated 2026-08-28 06:24:08 UTC by Nico là