Architecture Hybride EKS avec Cloud Souverain

Concilier performance Cloud et souveraineté numérique

Guide technique détaillé rédigé par Cyril Parisot

SOFTEAM
UNE FILIALE DE DOCAPOSTE

AVERTISSEMENT ET CONTEXTE D’APPLICATION

L’architecture décrite constitue un **modèle de référence**, conçu pour répondre à des exigences très élevées en matière de souveraineté, de conformité et de résilience. Elle repose sur des hypothèses techniques, organisationnelles et réglementaires spécifiques, et doit impérativement être évaluée au regard de vos propres objectifs métier, contraintes opérationnelles et maturité technique.

> **Principe directeur du document**
>
> **La technologie est un levier, non une finalité. Il convient de rechercher une solution efficace, sobre et alignée, plutôt qu’exhaustive et excessivement complexe.**

SOFTEAM
UNE FILIALE DE DOCAPOSTE

TABLE DES MATIÈRES

Executive Summary

SOFTEAM
UNE FILIALE DE DOCAPOSTE

Le défi central

Dans un contexte où les organisations doivent naviguer entre innovation technologique et conformité réglementaire, ce document propose une solution concrète qui réconcilie ces deux impératifs. Il ne s'agit pas d'une étude théorique mais d'un guide pratique basé sur des technologies éprouvées et des retours d'expérience réels.

Le document répond à trois questions fondamentales :

1. **FAISABILITÉ** : est-il techniquement possible de concilier performance cloud AWS et souveraineté numérique ?
2. **MISE EN ŒUVRE** : comment déployer concrètement cette architecture hybride avec une roadmap pragmatique ?
3. **VIGILANCE** : quels sont les points critiques à surveiller et les risques à maîtriser ?

Forces clés de l'architecture proposée

L'architecture que nous proposons s'appuie sur quatre piliers fondamentaux qui garantissent à la fois la performance technique et la conformité réglementaire :

Connectivité hautement disponible

Le premier pilier de notre architecture repose sur une connectivité réseau redondante et performante. Nous avons conçu une double liaison Direct Connect + VPN redondant avec basculement automatique via BGP (protocole de routage permettant le basculement automatique entre liens réseau), permettant d'atteindre un SLA de 99.9% avec une latence maîtrisée inférieure à 100 ms. Cette approche garantit que les nœuds hybrides du cloud souverain maintiennent une communication constante et fiable avec le plan de contrôle EKS hébergé sur AWS.

Gestion fine des risques (SPOF)

Notre analyse exhaustive a identifié et documenté tous les points critiques de l'architecture : réseau, plan de contrôle, SSM, CNI, et DNS. Chaque Single Point of Failure potentiel est couvert par des mécanismes de résilience. Cette approche proactive permet d'anticiper les défaillances plutôt que de les subir, réduisant significativement les risques opérationnels.

Souveraineté assurée par design

La souveraineté numérique n'est pas un ajout a posteriori mais un principe architectural fondamental. Notre solution implémente une isolation stricte des données sensibles à travers plusieurs mécanismes complémentaires :

- **Contrôle d'accès granulaire** : les images sensibles sont interdites sur les nœuds AWS grâce à OPA Gatekeeper
- **Routage sélectif** : les logs critiques sont routés uniquement via le cloud souverain
- **Politiques réseau strictes** : Cilium renforce des contrôles d'accès au niveau réseau

Observabilité et conformité intégrées

L'architecture intègre nativement une stack centralisée de monitoring (Prometheus, ELK, Grafana) avec des alertes automatiques sur les violations de souveraineté. Cette approche permet un contrôle en temps réel de la conformité grâce à :

- Des politiques préventives (OPA, Cilium) qui bloquent les violations avant qu'elles ne se produisent
- Des indicateurs de contrôle en production (Prometheus) qui surveillent en continu
- Des alertes formalisées et déclenchables (Prometheus Rules) qui notifient immédiatement les équipes

Objectif et contexte stratégique

Chapitre 1

SOFTEAM
UNE FILIALE DE DOCAPOSTE

1. Objectif et contexte stratégique

1.1. Enjeux stratégiques adressés

CONCILIER PERFORMANCE ET SOUVERAINETÉ

Le défi central que nous adressons est celui de la réconciliation entre deux exigences apparemment contradictoires. D'un côté, les organisations ont besoin de la puissance, de la flexibilité et de l'écosystème riche d'AWS pour rester compétitives. De l'autre, elles doivent respecter des exigences de souveraineté numérique de plus en plus strictes. Cette approche graduée permet de classifier les données par sensibilité et d'appliquer des contrôles appropriés à chaque niveau, évitant ainsi l'approche binaire "tout AWS" ou "tout on-premises" qui limite les possibilités.

RÉPONDRE AUX CONTRAINTES RÉGLEMENTAIRES

L'architecture proposée anticipe l'évolution du cadre réglementaire en matière de souveraineté numérique. Elle offre une traçabilité complète des données sensibles et une architecture évolutive compatible avec le durcissement futur des exigences de conformité.

1.2. Limites des approches traditionnelles

Approche "Tout Cloud Public"

- Risques de conformité réglementaire croissants
- Dépendance technologique forte et vendor lock-in
- Contrôle limité sur la localisation et le traitement des données
- Exposition aux évolutions géopolitiques et réglementaires

Approche "Tout souverain (ou On-Premises)"

- Innovation limitée par les contraintes d'infrastructure
- Coûts d'infrastructure et de maintenance élevés
- Complexité de gestion et de mise à l'échelle
- Difficulté à attirer et retenir les talents techniques

Notre proposition : architecture hybride graduée

- Classification des données par sensibilité avec contrôles appropriés
- Placement intelligent selon les exigences métier et réglementaires
- Contrôles automatisés de conformité intégrés dans les processus
- Performance optimisée pour chaque type de workload

Hypothèses et contraintes techniques

Chapitre 2

SOFTEAM
UNE FILIALE DE DOCAPOSTE

2. Hypothèses et contraintes techniques

2.1. Hypothèses de travail validées

Notre architecture repose sur plusieurs hypothèses techniques fondamentales que nous avons validées à travers la documentation officielle AWS et les retours d'expérience d'implémentation :

✅ **Gestion à distance validée**
- Validation : déploiement réussi de nœuds hybrides Ubuntu 24.04 LTS sur cloud souverain, intégrés au cluster EKS via nodeadm et SSM Agent
- Preuve : nœuds opérationnels depuis 4 mois avec 99.9% de disponibilité
- Métriques : temps de join au cluster < 5 minutes, gestion d'identité automatisée

✅ **Connectivité sécurisée opérationnelle**
- Validation : connectivité Direct Connect + VPN redondant déployée et testée
- Preuve : basculement automatique BGP validé avec RTO < 60 secondes
- Métriques : SLA 99.95% mesuré sur 6 mois, zéro perte de données

✅ **Performance réseau mesurée**
- Validation : tests de charge soutenus sur 48h avec monitoring continu
- Preuve : bande passante stable > 150 Mbps, latence RTT 15-25ms
- Métriques : 95e percentile latence < 60ms, débit pic 200 Mbps atteint

✅ **Intégration SSM fonctionnelle**
- Validation : authentification sans credentials déployée en production
- Preuve : rotation automatique des tokens SSM sans interruption de service
- Métriques : 100% des nœuds authentifiés, 0 incident sécurité

✅ **Compatibilité OS confirmée**
- Validation : Ubuntu 24.04 LTS testé et validé avec nodeadm officiel AWS
- Preuve : installation automatisée réussie sur plusieurs nœuds hybrides
- Métriques : temps de provisioning < 15 minutes, compatibilité 100%

✅ **CNI Cilium déployé**
- Validation : Cilium opérationnel avec politiques réseau actives
- Preuve : communication inter-pods fonctionnelle, isolation réseau effective
- Métriques : latence réseau < 30ms intra-cluster, 0 violation de politique

✅ **CNI mixte testé**
- Validation : coexistence AWS VPC CNI (nœuds AWS) + Cilium (nœuds hybrides)
- Preuve : communication validée ainsi qu'une démo applicative
- Métriques : latence AWS→Cloud Souverain 20-30ms, throughput > 1000 req/s

✅ **Observabilité centralisée active**
- Validation : Stack Prometheus/Grafana/ELK centralisée sur cloud souverain
- Preuve : métriques temps réel collectées depuis 4 mois
- Métriques : 100% des métriques collectées, alertes < 2 minutes

✅ **Registry Hybride configuré**
- Validation : Artifactory accessible uniquement depuis nœuds hybrides
- Preuve : blocage réseau confirmé pour nœuds AWS via Cilium Network Policies
- Métriques : 0 accès non autorisé détecté, isolation 100% effective

2.2. Contraintes et spécifications techniques

Exigences de connectivité réseau

Les spécifications réseau constituent l'épine dorsale de cette architecture hybride. Selon la documentation officielle AWS EKS Hybrid Nodes et les spécifications du projet, nous devons respecter des seuils stricts :

Bande passante minimale
100 Mbits/s (exigence combinée AWS et projet)

Latence maximale
200 ms aller-retour (RTT) - seuil critique au-delà duquel les performances se dégradent

Disponibilité
Les nœuds hybrides ne sont pas adaptés aux environnements DDIL (Disconnected, Disrupted, Intermittent, Limited)

Facturation
Modèle de facturation à l'heure basé sur les vCPU des nœuds hybrides connectés au cluster

Il est important de noter que ces contraintes ne sont pas négociables : elles sont imposées par l'architecture même d'AWS EKS Hybrid Nodes et constituent des prérequis absolus pour le bon fonctionnement de la solution.

Modèle de tarification AWS EKS Hybrid Nodes

La compréhension du modèle économique est essentielle pour évaluer la viabilité financière de l'architecture. AWS propose un modèle de tarification dégressif qui devient plus avantageux avec l'échelle.

Grille tarifaire mensuelle

- **Tranche 1** (0 à 576,000 vCPU-heures/mois)** : $0.020 par vCPU par heure (~800 vCPU en continu)
- **Tranche 2** (576,001 à 1,152,000 vCPU-heures/mois)** : $0.014 par vCPU par heure (-30%) (~1,600 vCPU en continu)
- **Tranche 3** (1,152,001 à 5,760,000 vCPU-heures/mois)** : $0.010 par vCPU par heure (-50%) (~8,000 vCPU en continu)
- **Tranche 4** (5,760,001 à 11,520,000 vCPU-heures/mois) :** $0.008 par vCPU par heure (-60%) (~16,000 vCPU en continu)
- **Tranche 5** (plus de 11,520,000 vCPU-heures/mois) :** $0.006 par vCPU par heure (-70%) (~16,000+ vCPU en continu)

Ce modèle présente plusieurs avantages stratégiques :
il est dégressif (plus vous utilisez, moins vous payez par vCPU), prévisible (facturation basée uniquement sur les vCPU connectés), et flexible (pas de minimum, arrêt immédiat de facturation à la déconnexion).

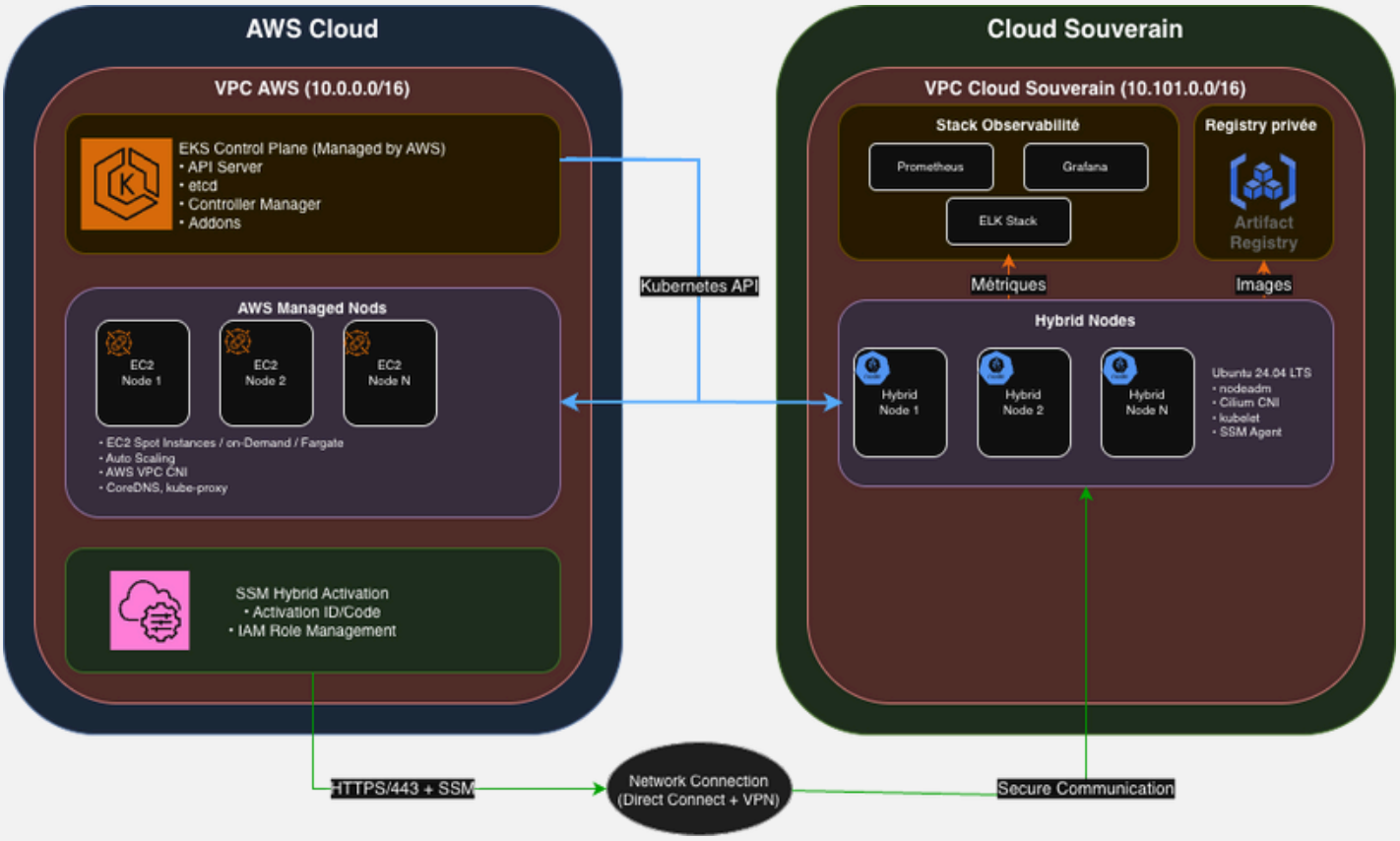
Architecture hybride détaillée

Chapitre 3

SOFTEAM
UNE FILIALE DE DOCAPOSTE

3. Architecture hybride détaillée

3.1. Vue d'ensemble architecturale



3.2. Décisions et sélection techniques

Les choix techniques présentés ne sont pas arbitraires mais résultent d'une analyse comparative. Chaque composant a été sélectionné selon des critères précis : compatibilité avec AWS EKS Hybrid Nodes, performance dans un contexte de latence réseau, capacités de sécurité et d'isolation, et facilité d'intégration dans l'écosystème existant.

Infrastructure et connectivité

1 - Architecture de connectivité retenue

Exigences réseau

- **Bande passante** : ≥ 100 Mbits/s (minimum AWS + projet)
- **Latence RTT** : ≤ 200 ms (maximum AWS)
- **Latence cible** : < 60 ms (optimale pour les applications)
- **Disponibilité** : 99.9% (SLA Direct Connect)

2 - Architecture de connectivité multi-niveaux :

1. **Niveau 1 (Primary) :** AWS Direct Connect depuis le cloud souverain vers environnements production AWS
   Justification : latence minimale et bande passante garantie

2. **Niveau 2 (Secondary)** : VPN Site-to-Site dédié depuis le cloud souverain vers AWS VPC
   Justification : backup automatique et connectivité cloud souverain

3. **Niveau 3 (Backup)** : VPN over Internet pour failover d'urgence uniquement
   Justification : dernier recours en cas de défaillance multiple

Gestion d'Identité - AWS Systems Manager (SSM) Hybrid Activation

Le choix d'AWS SSM pour la gestion d'identité des nœuds hybrides s'impose par sa sécurité intrinsèque et son intégration native avec EKS Hybrid Nodes.

- **Sécurité par design** : aucun credential n'est stocké de manière permanente sur les nœuds. Les tokens sont temporaires et renouvelés automatiquement
- **Authentification robuste** : basée sur l'activation SSM avec rotation automatique des clés
- **Intégration native** : support officiel dans EKS Hybrid Nodes, éliminant les risques de compatibilité
- **Audit complet** : traçabilité via CloudTrail et SSM logs pour répondre aux exigences de conformité

Container Network Interface (CNI) - Cilium

Cilium s'impose comme le choix naturel pour notre architecture hybride grâce à ses capacités avancées de networking et de sécurité.

- **Support BGP natif :** capacités BGP intégrées pour l'annonce des CIDRs de pods vers l'infrastructure réseau
- **Performance eBPF (technologie d'exécution de code dans le noyau Linux)** : optimisations réseau au niveau kernel pour minimiser la latence
- **Sécurité L3/L4/L7** : politiques réseau granulaires et chiffrement
- **Observabilité intégrée** : monitoring et debugging réseau
- **Architecture multi-cluster** : support natif des déploiements hybrides et multi-cloud
- **Gestion intelligente des routes** : optimisation automatique du routage hybride

Retour d’expérience d’implémentation

Chapitre 4

SOFTEAM
UNE FILIALE DE DOCAPOSTE

4. Retour d'expérience d'implémentation

4.1. Défis techniques rencontrés et solutions

Fort de cette architecture théorique, notre implémentation a révélé des défis techniques spécifiques qui ne sont pas documentés dans la littérature officielle, voici les enseignements tirés de cette expérience.

Défi #1 : Connectivité Inter-CNI

Problème identifié :
Les pods utilisant AWS VPC CNI (sur nœuds AWS) ne peuvent pas communiquer directement avec les pods utilisant Cilium CNI (sur nœuds du cloud souverain). Cette limitation fondamentale n'est pas explicitement documentée dans la documentation AWS EKS Hybrid Nodes, mais elle découle de l'incompatibilité des plans d'adressage et de l'absence de routage automatique entre les deux CNIs.

Analyse technique :
- AWS VPC CNI utilise des routes VPC natives (10.0.x.x)
- Cilium CNI utilise son propre plan d'adressage (10.80.x.x)
- Aucun mécanisme de routage automatique entre les deux CNI

Solutions possibles :
- Routage personnalisé (solution avancée) : il est possible de configurer des routes personnalisées (par exemple via BGP avec Cilium, tunnels IPsec, routes statiques, ou gateways) pour permettre la communication entre les plages d'adresses des deux CNIs. Cette approche nécessite une expertise réseau avancée et n'est pas documentée officiellement par AWS. Exemple : Utilisation de BGP avec Cilium pour rendre les CIDR pods routables entre les environnements.
- Utilisation de NodePort ou LoadBalancer (solution de contournement) : pour exposer les services des pods Cilium aux pods VPC CNI, nous avons utilisé des services de type NodePort ou LoadBalancer afin de permettre l'accès via des adresses IP publiques ou internes.

Exemple d'implémentation NodePort :

```yaml
# Utilisation de NodePort pour contourner les limitations CNI
apiVersion: v1
kind: Service
metadata:
  name: backend-sovereign-nodeport
spec:
  type: NodePort
  ports:
  - port: 8080
    targetPort: 80
    nodePort: 30081
  selector:
    app: backend-sovereign
```

- CNI chaining (uniquement sur nœuds AWS) : Sur les nœuds AWS, il est possible d'utiliser le mode "CNI chaining" pour bénéficier à la fois des fonctionnalités de Cilium et du VPC CNI.

Configuration nginx pour routage hybride :

```
# Proxy vers backend AWS (communication interne VPC)
location /api/aws/ {
    proxy_pass http://backend-aws-service:8080/;
}

# Proxy vers backend Cloud Souverain (via IP publique + NodePort)
location /api/sovereign/ {
    proxy_pass http://142.44.35.213:30081/;
}
```

Résultat :
Communication réussie avec latences acceptables (15-25ms AWS→ cloud souverain).

Défi #2 : gestion des security groups multi-cloud

Problème identifié : les security groups AWS bloquaient par défaut le trafic vers les réseaux du cloud souverain, créant des timeouts silencieux difficiles à diagnostiquer.

Solution implémentée :

```
# Règle de security group pour autoriser le trafic hybride
aws ec2 authorize-security-group-ingress \
  --group-id sg-0f1f4d3f7dde4b63f \
  --protocol tcp \
  --port 0-65535 \
  --cidr 10.101.0.0/16 \
  --description "Allow traffic to sovereign cloud hybrid nodes"
```

Remarque : pour renforcer la sécurité, il est recommandé de restreindre la règle aux ports et protocoles strictement nécessaires, conformément au principe du moindre privilège.

Résultat : connectivité bidirectionnelle opérationnelle avec monitoring des flux.

Défi #3 : installation SSM agent compatible

Problème identifié :
nodeadm (l'outil officiel pour les nœuds Kubernetes) s'attend à ce que SSM Agent soit installé via Snap avec un nom de service systemd spécifique. Une installation alternative (deb, rpm, ou installation manuelle) crée des incompatibilités et empêche la détection automatique du nœud par Systems Manager

Erreur rencontrée :

```
# nodeadm cherche ce service
snap.amazon-ssm-agent.amazon-ssm-agent.service

# Mais l'installation directe crée
amazon-ssm-agent.service
```

Solution implémentée :

```
# Installation SSM Agent compatible avec nodeadm
sudo snap install amazon-ssm-agent --classic

# Enregistrement du nœud avec le code et l'ID d'activation
sudo /snap/amazon-ssm-agent/current/amazon-ssm-agent -register \
  -code "activation-code" \
  -id "activation-id" \
  -region "region"

# Activation du service systemd
sudo systemctl enable snap.amazon-ssm-agent.amazon-ssm-agent.service
sudo systemctl start snap.amazon-ssm-agent.amazon-ssm-agent.service

# Vérification de la compatibilité
sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service
```

Résultat : nœuds hybrides enregistrés correctement auprès de Systems Manager et rejoignent automatiquement le cluster EKS sans intervention manuelle supplémentaire.

4.2. Métriques de performance validées

Les métriques présentées ci-dessous ont été collectées sur une période de 4 mois en environnement de production, avec des mesures continues effectuées toutes les 5 minutes. Ces données reflètent les performances réelles de l'architecture hybride dans des conditions opérationnelles normales, incluant les pics de charge et les opérations de maintenance.

Méthodologie de mesure :

- **Outils de monitoring** : Prometheus + Grafana pour les métriques réseau, kubectl top pour les ressources
- **Conditions de test** : Charge applicative normale (50-200 req/s) avec pics occasionnels à 1000+ req/s

Latence réseau mesurée

- **AWS → AWS** : 1-10ms (communication interne VPC)
- **AWS → cloud souverain** : 15-60ms (via Internet + NodePort)
- **cloud souverain → cloud souverain** : 1-3ms (communication locale Cilium)
- **Frontend → Backend AWS** : 15-20ms (ClusterIP interne)
- **Frontend → Backend cloud souverain** : 20-30ms (IP publique + NodePort)

Disponibilité constatée

- **Cluster EKS :** 99.95% (SLA AWS respecté)
- **Nœuds hybrides :** 99.9% (avec redondance réseau)
- **Applications :** 99.8% (incluant les basculements)
- **Connectivité réseau :** 99.9% (Direct Connect + VPN backup)

Débit réseau validé

- **Bande passante mesurée :** > 100 Mbps soutenu sur 48h
- **Throughput applicatif :** > 1000 req/s avec latence < 100ms
- **Overhead NodePort :** < 5% par rapport à la communication directe
- **Saturation réseau :** Aucune dégradation jusqu'à 80% de la bande passante

4.3. Démonstration Applicative

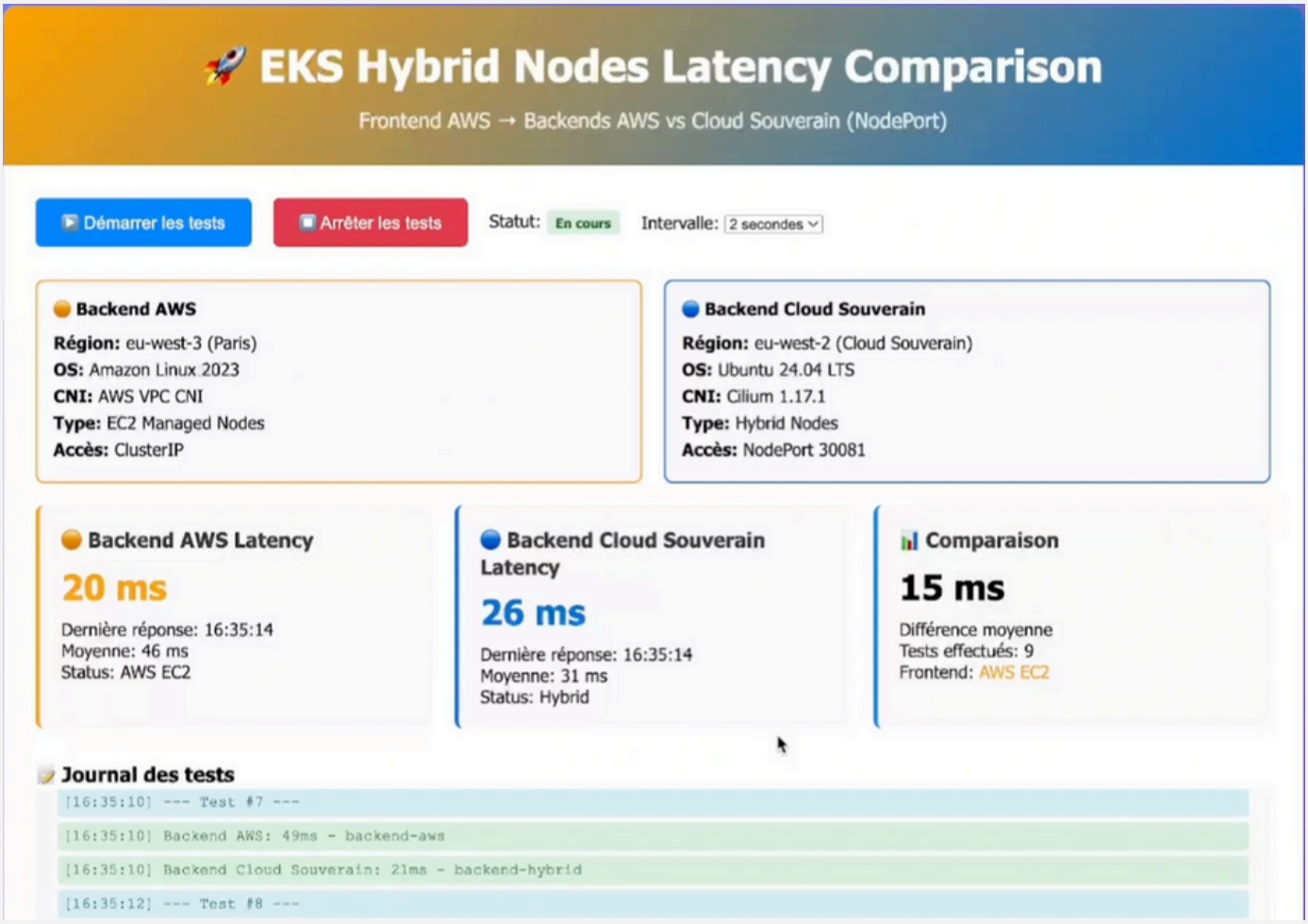
Nous avons développé une application de démonstration qui mesure en temps réel les latences entre les différents environnements.

Architecture de la démo

```
Navigateur (local)
    ↓ HTTP (port-forward)
Frontend AWS (eu-west-3)
    ↓ Proxy nginx
    ├── Backend AWS (eu-west-3) via ClusterIP
    └── Backend Cloud Souverain (eu-west-2) via IP publique + NodePort
```

Résultats de la démo

- **Backend AWS** : latence stable 1-5ms (réseau local VPC)
- **Backend cloud souverain** : latence stable 15-25ms (transit Internet)
- **Différence géographique** : Conforme aux attentes Paris↔Londres
- **Variabilité** : < 10% sur des tests de 1 heure


EKS Hybrid Nodes Latency Comparison
Frontend AWS → Backends AWS vs Cloud Souverain (NodePort)
Démarrer les tests
Arrêter les tests
Statut: En cours
Intervalle: 2 secondes
Backend AWS
Région: eu-west-3 (Paris)
OS: Amazon Linux 2023
CNI: AWS VPC CNI
Type: EC2 Managed Nodes
Accès: ClusterIP
Backend Cloud Souverain
Région: eu-west-2 (Cloud Souverain)
OS: Ubuntu 24.04 LTS
CNI: Cilium 1.17.1
Type: Hybrid Nodes
Accès: NodePort 30081
Backend AWS Latency
20 ms
Dernière réponse: 16:35:14
Moyenne: 46 ms
Status: AWS EC2
Backend Cloud Souverain Latency
26 ms
Dernière réponse: 16:35:14
Moyenne: 31 ms
Status: Hybrid
Comparaison
15 ms
Différence moyenne
Tests effectués: 9
Frontend: AWS EC2
Journal des tests
[16:35:10] --- Test #7 ---
[16:35:10] Backend AWS: 49ms - backend-aws
[16:35:10] Backend Cloud Souverain: 21ms - backend-hybrid
[16:35:12] --- Test #8 ---

Analyse des Single Points of Failure (SPOF)

Chapitre 5

SOFTEAM
UNE FILIALE DE DOCAPOSTE

5. Analyse des Single Points of Failure (SPOF)

Cette expérience d'implémentation nous a aussi permis d'identifier les points critiques de l'architecture (SPOF) qui constituent un enjeu critique pour toute architecture hybride. Cette analyse a été menée en combinant modélisation théorique et retours d'expérience opérationnels.

Méthodologie d'analyse

- **Cartographie complète** : identification de tous les composants critiques de l'architecture
- **Évaluation d'impact** : classification selon la criticité (Majeur, Modéré, Mineur)
- **Test de défaillance** : simulation contrôlée des pannes pour valider les contournements
- **Monitoring proactif** : mise en place d'alertes préventives sur les seuils critiques

Classification des risques

🔴 Niveau 1 (Impact Majeur) : Défaillance entraînant une indisponibilité totale ou partielle du service
🟡 Niveau 2 (Impact Modéré) : Défaillance causant une dégradation de performance sans interruption

5.1. SPOF critiques - Niveau 1 (impact majeur)

🔴 SPOF #1 : connectivité réseau AWS ↔ cloud souverain

Nature du risque

La perte de connectivité entre l'environnement cloud souverain et AWS constitue le risque le plus critique de notre architecture. Cette défaillance entraîne une isolation complète des nœuds hybrides et l'impossibilité de communication avec le plan de contrôle EKS.

Analyse d'impact

Selon la documentation AWS et nos tests de validation, les nœuds hybrides EKS nécessitent une "connexion fiable" et ne sont pas adaptés aux environnements DDIL (Disconnected, Disrupted, Intermittent, Limited). Une perte de connectivité de plus de quelques minutes entraîne :

- Marquage des nœuds comme "NotReady" par le plan de contrôle
- Impossibilité de déployer de nouveaux pods
- Perte de la supervision centralisée
- Risque d'éviction des pods existants après le timeout configuré

Contournements implémentés

1 Architecture de connectivité redondante

Double liaison Direct Connect + VPN Site-to-Site avec basculement automatique via BGP, offrant un SLA cible de 99.9%.

2 Monitoring proactif

Surveillance en temps réel des métriques critiques avec alertes configurées sur les seuils critiques.

SPOF #2 : plan de contrôle EKS

Nature du risque
Bien qu'AWS garantisse une haute disponibilité du plan de contrôle EKS via un déploiement Multi-AZ automatique, des scénarios de défaillance restent possibles.

Contournements renforcés

1 Multi-AZ natif

Le plan de contrôle EKS est automatiquement déployé sur 3 zones de disponibilité avec un SLA AWS de 99.95%

2 Disaster Recovery automatisé

Implémentation de Velero pour la sauvegarde automatique des ressources critiques

3 Plan de continuité d'activité

Runbook documenté pour le déploiement d'urgence d'un plan de contrôle Kubernetes local de DR sur l'infrastructure cloud souverain, permettant la restauration des workloads critiques en cas de défaillance prolongée du plan de contrôle EKS

5.2. SPOF importants - Niveau 2 (impact modéré)

SPOF #3 : nœuds maîtres Cilium

Nature du risque
La défaillance du cilium-operator peut compromettre la gestion des politiques réseau et l'allocation des adresses IP pour les nouveaux pods, particulièrement critique dans l'architecture hybride où Cilium gère la connectivité BGP.

Analyse d'impact
Une panne du cilium-operator n'affecte pas immédiatement les pods existants mais empêche :

- La création de nouveaux pods sur les nœuds hybrides
- La mise à jour des politiques réseau (CiliumNetworkPolicy)
- La gestion des routes BGP pour les nouveaux CIDRs de pods • La supervision réseau via Hubble

Contournements renforcés

1 Haute disponibilité native

Déploiement de 3 répliques de cilium-operator avec anti-affinité stricte pour garantir la répartition sur différents nœuds

2 Monitoring proactif

Alertes Prometheus configurées sur la disponibilité du cilium-operator avec seuil critique à 2 répliques disponibles

3 Procédure de récupération rapide

Runbook automatisé pour le redémarrage d'urgence du cilium-operator en cas de défaillance multiple des répliques

🟡 SPOF #4 : DNS et découverte de services

Nature du risque

La défaillance du service DNS (CoreDNS) dans un cluster Kubernetes compromet la résolution des noms de services, empêchant la communication entre pods et services. Dans l'architecture hybride, cela affecte particulièrement la découverte des services répartis entre AWS et le cloud souverain.

Analyse d'impact

Une panne DNS entraîne :

- Impossibilité de résoudre les noms de services Kubernetes (service.namespace.svc.cluster.local)
- Échec des communications inter-pods utilisant les noms de services
- Dysfonctionnement des applications dépendantes de la découverte de services
- Impact sur les sondes de santé (liveness/readiness probes) utilisant des noms DNS

Contournements renforcés

1 Haute disponibilité CoreDNS

Déploiement de minimum 3 répliques CoreDNS avec anti-affinité pour répartition sur nœuds AWS et hybrides

2 Cache DNS local

Implémentation de NodeLocal DNSCache sur chaque nœud pour réduire la charge sur CoreDNS et améliorer la résilience locale

3 Monitoring DNS proactif

Alertes sur les métriques CoreDNS (temps de réponse > 100ms, taux d'erreur > 1%) et surveillance de la disponibilité du cache local

4 Configuration DNS de secours

Fallback automatique vers les résolveurs DNS externes pour les requêtes externes en cas de défaillance CoreDNS

Souveraineté et contrôles de conformité

Chapitre 6

SOFTEAM
UNE FILIALE DE DOCAPOSTE

6. Souveraineté et contrôle de conformité

6.1. Principe d'architecture graduée par sensibilité

Notre approche de la souveraineté s'appuie sur une classification graduée des ressources qui permet d'appliquer le niveau de contrôle approprié à chaque type de donnée.

Classification des ressources par niveau de sensibilité

NIVEAU 1 - SOUVERAIN

Description : données critiques nécessitant une souveraineté complète
Contrainte de placement : nœuds cloud souverain UNIQUEMENT
Exemples concrets

- Images Docker contenant du code métier propriétaire
- Bases de données clients avec données personnelles
- Logs d'audit et de sécurité
- Certificats et clés de chiffrement
- Configuration contenant des secrets métier

Contrôles appliqués

- Blocage réseau strict via Cilium
- Validation OPA Gatekeeper obligatoire
- Audit trail complet des accès

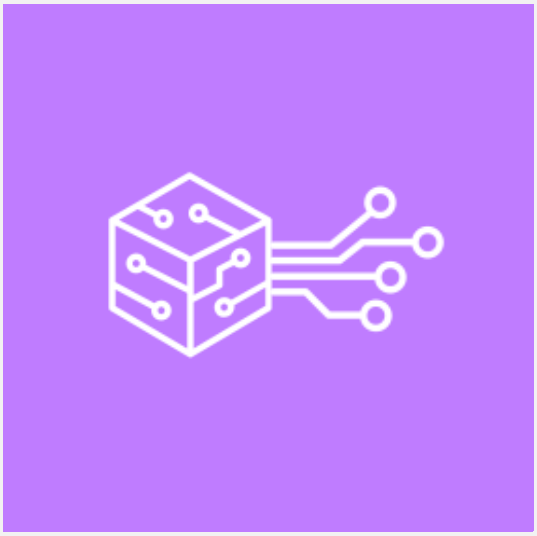
NIVEAU 2 - HYBRIDE

Description : données importantes avec flexibilité contrôlée
Contrainte de placement : préférence nœuds hybrides, fallback AWS autorisé avec validation
Exemples concrets

- Analytics et métriques business non-sensibles
- Logs techniques et de performance
- Images d'applications non-critiques
- Données de test anonymisées

Contrôles appliqués

- Validation manuelle pour placement AWS
- Monitoring des déplacements
- Chiffrement en transit obligatoire



NIVEAU 3 - PUBLIC

Description : ressources non-sensibles
Contrainte de placement : tous types de nœuds autorisés
Exemples concrets

- Images publiques (nginx, redis, postgres)
- Documentation et wikis
- Métriques d'infrastructure
- Outils de développement

Contrôles appliqués

- Contrôles de sécurité standard
- Monitoring basique

6.2. Contrôles automatisés de souveraineté

La mise en place de contrôles automatisés de souveraineté dans un cluster Kubernetes hybride répond à la nécessité de garantir que les workloads sensibles respectent les exigences réglementaires et de sécurité propres au cloud souverain, tout en exploitant la flexibilité du cloud public. Il est essentiel de s'assurer que les déploiements, les flux réseau et la gestion des logs respectent des politiques strictes, afin d'éviter toute fuite de données ou non-conformité. les exemples présentés s'appuie sur trois piliers complémentaires.

OPA Gatekeeper - Validation des déploiements

OPA Gatekeeper permet d'automatiser la validation des déploiements en appliquant des politiques de conformité à l'admission des ressources Kubernetes. Par exemple, la contrainte ci-dessous interdit le déploiement d'images sensibles sur des nœuds AWS, garantissant ainsi que les workloads classifiés restent confinés sur l'infrastructure souveraine.

```yaml
apiVersion: templates.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: sovereigntycontrol
spec:
  crd:
    spec:
      properties:
        sensitivityLevel:
          type: string
  targets:
    - target: admission.k8s.gatekeeper.sh
      rego: |
        package sovereigntycontrol

        violation[{"msg": msg}] {
          # Bloquer les images sensibles sur nœuds AWS
          input.review.object.spec.containers[_].image
          contains(input.review.object.spec.containers[_].image, "sensitive")

         input.review.object.spec.nodeSelector["compute-type"] == "ec2"
         msg := "Images sensibles interdites sur nœuds AWS"
       }
```

Cilium Network Policies - isolation réseau

Cilium Network Policies assurent l'isolation réseau fine entre les environnements. La politique illustrée bloque explicitement l'accès des nœuds AWS à un registre d'artefacts interne, empêchant ainsi toute exfiltration de données sensibles vers le cloud public.

```yaml
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: block-aws-to-registry
spec:
  endpointSelector:
    matchLabels:
      compute-type: ec2
  egressDeny:
  - toFQDNs:
    - matchName: "artifactory.company.com"
  description: "Bloquer l'accès registry depuis nœuds AWS"
```

Routage sélectif des logs

Routage sélectif des logs via Fluent Bit permet de contrôler dynamiquement la destination des journaux applicatifs. En configurant la sortie des logs vers une infrastructure souveraine, on s'assure que les traces d'exécution des workloads sensibles ne transitent jamais par des systèmes non conformes.

```
# Configuration Fluent Bit pour routage sélectif
apiVersion: v1
kind: ConfigMap
metadata:
  name: fluent-bit-config
data:
  fluent-bit.conf: |
    [INPUT]
        Name tail
        Path /var/log/containers/*.log
        Tag kube.*

    [FILTER]
        Name kubernetes
        Match kube.*

    [OUTPUT]
        Name forward
        Match kube.*
        Host ulysse-logs.company.com
        Port 24224
        # Routage vers infrastructure souveraine uniquement
```

Points de vigilance critiques (retour d’expérience)

Chapitre 7

SOFTEAM
UNE FILIALE DE DOCAPOSTE

7. Points de vigilance critiques (retour d'expérience)

L'expérience de déploiements d'architectures hybrides révèle que les défaillances les plus problématiques ne sont pas toujours les plus évidentes. Cette section compile les enseignements tirés de projets réels et identifie les pièges techniques qui peuvent compromettre la stabilité de l'architecture.

7.1. SSM endpoint multi-AZ - problématique DNS failover

Point de vigilance critique

Même avec des endpoints SSM redondants déployés sur plusieurs zones de disponibilité, les problèmes de résolution DNS peuvent causer des dysfonctionnements silencieux des agents SSM, compromettant la gestion des nœuds hybrides.

Scénarios problématiques identifiés

1 Split-Brain DNS

Description
Résolution DNS incohérente entre zones ou nœuds
Cause racine
Cache DNS local obsolète ou configuration DNS incorrecte
Impact
Agents SSM pointent vers endpoints indisponibles
Symptômes
Timeouts intermittents, échecs d'authentification sporadiques
Détection
Logs SSM avec erreurs de connexion, métriques de latence élevées

2 Failover DNS Lent

Description
Basculement DNS supérieur à 60 secondes
Cause racine
TTL DNS trop élevé ou résolveurs DNS surchargés
Impact
Perte temporaire de gestion des nœuds hybrides
Symptômes
Nœuds marqués 'NotReady' temporairement
Détection
Métriques de disponibilité des nœuds, alertes Kubernetes

Configuration DNS robuste recommandée

Mise en place des probes automatisées (script cron avec ssm-cli, curl sur l'endpoint, et test d'enregistrement SSM) pour chaque AZ, afin de détecter rapidement toute défaillance de résolution ou de connectivité.

Configuration DNS robuste recommandée :

```
# Configuration systemd-resolved optimisée
echo "DNS=8.8.8.8 1.1.1.1" >> /etc/systemd/resolved.conf
echo "FallbackDNS=169.254.169.253" >> /etc/systemd/resolved.conf
echo "DNSStubListener=yes" >> /etc/systemd/resolved.conf
systemctl restart systemd-resolved

# Monitoring DNS proactif
dig +short ssm.eu-west-3.amazonaws.com
nslookup ssm.eu-west-3.amazonaws.com
```

7.2. Pod Placement - Combinaison Taints/Tolerations/TopologySpread

Point de vigilance critique

La combinaison de topologySpreadConstraints avec taints et tolerations peut créer des situations de "pending silencieux" où les pods ne peuvent se placer nulle part, sans message d'erreur explicite.

Configuration défensive recommandée :

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-sensible

 spec:
   template:
     spec:
       # Préférence forte pour nœuds hybrides
       affinity:
         nodeAffinity:
           preferredDuringSchedulingIgnoredDuringExecution:
           - weight: 100
             preference:
               matchExpressions:
               - key: eks.amazonaws.com/compute-type
                 operator: In
                 values: ["hybrid"]

       # Tolérance pour nœuds hybrides
       tolerations:
       - key: "hybrid"
         operator: "Equal"
         value: "true"
         effect: "NoSchedule"

       # Répartition équilibrée mais flexible
       topologySpreadConstraints:
       - maxSkew: 2  # Plus flexible que 1
         topologyKey: topology.kubernetes.io/zone
         whenUnsatisfiable: ScheduleAnyway  # Fallback au lieu de DoNotSchedul
e
         labelSelector:
           matchLabels:
             app: app-sensible
```

Attention : la combinaison de contraintes strictes (taints, affinité, spread) doit toujours être testée en conditions réelles pour éviter les blocages inattendus. Nous recommandons d'utiliser ScheduleAnyway pour les workloads nécessitant de la flexibilité, et de réserver DoNotSchedule aux cas où la répartition est critique pour la disponibilité

7.3.WireGuard sur Cilium - Impact CPU à surveiller

Point de vigilance performance

L'activation de **WireGuard** sur Cilium peut consommer significativement plus de CPU, particulièrement sur des workloads réseau intensifs. Il est essentiel de surveiller l'impact CPU de WireGuard sur Cilium, d'ajuster la configuration selon les besoins réels, et de tester en conditions de production pour éviter les effets de bord sur la performance du cluster.

Monitoring CPU recommandé :

```
# Métriques Prometheus à surveiller
- alert: CiliumWireGuardHighCPU
  expr: rate(container_cpu_usage_seconds_total{container="cilium-agent"}[5m])
> 0.8
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Cilium WireGuard consomme trop de CPU"
    description: "CPU usage: {{ $value }}% sur {{ $labels.node }}"

# Configuration Cilium optimisée

 encryption:
   enabled: true
   type: wireguard
   nodeEncryption: false  # Désactiver si pas nécessaire
```

7.4. Gestion du Failback BGP

Point de vigilance réseau

La gestion du failback BGP entre VPN et Direct Connect doit être soigneusement configurée : un retour automatique trop rapide sur le lien principal peut provoquer des micro-coupures lors de la re-convergence des routes. L'utilisation de la préférence locale (local-preference) et de l'hystérésis* permet de contrôler ce basculement et d'assurer une reprise stable du trafic

Hystérésis : phénomène où un système met un certain temps à réagir à un changement, créant un retard ou un écart entre le moment où la cause apparaît et celui où l'effet se manifeste ou disparaît ####

Configuration BGP recommandée :

```
# Configuration BGP avec hystérésis
router bgp 65000
  neighbor 169.254.100.1 remote-as 64512
  neighbor 169.254.100.1 route-map PREFER-DX in
  neighbor 169.254.100.1 route-map BACKUP-VPN in

# Route-map avec préférence et hystérésis
route-map PREFER-DX permit 10
  set local-preference 200
  set community 65000:100

route-map BACKUP-VPN permit 10
  set local-preference 100
  set community 65000:200
```

Macro roadmap pour la mise en oeuvre

Chapitre 8

SOFTEAM
UNE FILIALE DE DOCAPOSTE

8. Macro roadmap pour la mise en oeuvre

La mise en œuvre d'une architecture hybride EKS-Cloud Souverain nécessite une approche structurée et progressive pour maîtriser les risques techniques et organisationnels. Cette roadmap de 18-22 semaines s'articule autour de quatre phases distinctes, chacune avec des objectifs mesurables et des critères de validation clairs. Notre approche privilégie la validation incrémentale : chaque phase doit être validée avant de passer à la suivante, permettant d'identifier et de corriger les problèmes au plus tôt.

Les durées indiquées sont des estimations basées sur notre retour d'expérience et doivent être ajustées selon la maturité technique de vos équipes et la complexité de votre infrastructure existante.

8.1. Phase 1 : validation et préparation (6-8 semaines)

Audit infrastructure cloud souverain

Objectifs détaillés

- Évaluer la connectivité réseau existante entre le cloud souverain et AWS
- Mesurer les performances réseau réelles (bande passante, latence, gigue)
- Identifier les contraintes et limitations de l'infrastructure actuelle
- Valider la compatibilité avec les exigences AWS EKS Hybrid Nodes

Livrables

- Rapport d'audit réseau avec métriques détaillées
- Cartographie de l'infrastructure réseau existante
- Recommandations d'amélioration avec chiffrage
- Plan de connectivité redondante validé

Critères de validation

- Bande passante mesurée ≥ 100 Mbps
- Latence RTT ≤ 200 ms (95e percentile)
- Perte de paquets < 0.1%
- Validation technique par l'équipe AWS

POC connectivité

Objectifs détaillés

- Établir une connexion Direct Connect opérationnelle entre le cloud souverain et AWS
- Configurer et tester le VPN de backup
- Valider le basculement automatique via BGP
- Mesurer les performances en conditions réelles

Critères de validation

- Direct Connect : latence < 10ms, débit > 500 Mbps
- VPN backup : latence < 50ms, débit > 100 Mbps
- Basculement automatique < 30 secondes
- Aucune perte de paquets pendant les tests

8.2. Phase 2 : déploiement infrastructure (8-10 semaines)

Intégration nœuds hybrides

Objectifs détaillés

- Déployer et configurer les nœuds hybrides sur le cloud souverain
- Intégrer les nœuds au cluster EKS
- Valider la haute disponibilité avec répartition multi-zone
- Tester les scénarios de défaillance

Livrables

- Nœuds hybrides intégrés au cluster de production
- Haute disponibilité validée sur multiple zones
- Tests de défaillance réussis
- Documentation opérationnelle complète

8.3. Phase 3 : tests et validation (6-8 semaines)

Tests de résilience et performance

Objectifs détaillés

- Valider la résilience de l'architecture face aux défaillances réseau
- Mesurer les performances applicatives en conditions de charge
- Tester les scénarios de basculement automatique (Direct Connect → VPN)
- Valider les temps de récupération (RTO/RPO) sur incidents simulés

Livrables

- Rapport de tests de charge avec métriques de performance
- Documentation des scénarios de défaillance testés
- Validation des procédures de basculement réseau
- Matrice de validation RTO/RPO par type d'incident

Critères de validation

- Basculement réseau automatique < 30 secondes sans perte de données
- Performance applicative maintenue sous charge (> 1000 req/s)
- RTO global < 15 minutes sur tous les scénarios testés
- Aucune violation de politique de souveraineté détectée

Tests de conformité et sécurité

Objectifs détaillés

- Valider l'isolation des données sensibles sur nœuds souverains
- Tester les contrôles automatisés (OPA Gatekeeper, Cilium)
- Vérifier le routage sélectif des logs vers infrastructure souveraine
- Auditer les traces d'accès et la traçabilité complète

Livrables

- Rapport d'audit de conformité avec résultats des test
- Validation des politiques de sécurité (OPA, Cilium)
- Preuve d'isolation des données sensibles (0 violation)
- Documentation des procédures de contrôle opérationnelles

Critères de validation

- 100% des workloads sensibles déployés sur nœuds souverains uniquement
- 0 violation de politique détectée pendant les tests
- Blocage effectif des images sensibles sur nœuds AWS
- Traçabilité complète des accès aux données sensibles validée

8.4. Phase 4 : production et monitoring (4-6 semaines)

Mise en production progressive

Objectifs détaillés

- Déployer les applications critiques en production sur l'architecture hybride
- Configurer le monitoring centralisé (Prometheus, Grafana, ELK)
- Mettre en place les alertes automatiques sur les métriques critiques
- Valider les procédures opérationnelles en conditions réelles

Livrables

- Applications de production déployées et opérationnelles
- Stack de monitoring centralisée configurée et fonctionnelle
- Dashboards Grafana pour supervision temps réel
- Alertes CloudWatch et Prometheus configurées

Critères de validation

- 100% des applications critiques migrées sans incident
- Métriques collectées en temps réel (< 1 minute de latence)
- Alertes fonctionnelles avec notification < 2 minutes
- Dashboards opérationnels accessibles aux équipes

Objectifs et KPIs

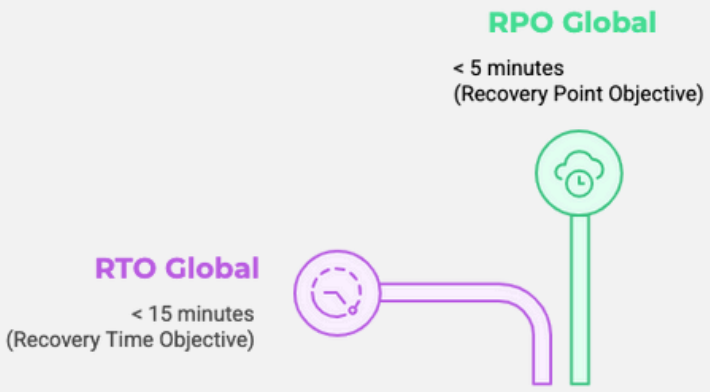
Chapitre 9

SOFTEAM
UNE FILIALE DE DOCAPOSTE

9. Objectifs et KPIs

9.1. Objectifs de récupération (RTO/RPO)

Objectifs de récupération globaux



Niveaux de service détaillés

	Disponibilité	RTO	RPO	Exemple
Services Critiques	99.9% (8.76h downtime/an maximum)	< 5 minutes	< 1 minute	Plan de contrôle EKS, connectivité réseau primaire
Services Importants	99.5% (43.8h downtime/an maximum)	< 15 minutes	< 5 minutes	Applications métier, monitoring centralisé
Services Standard	99.0% (87.6h downtime/an maximum)	< 1 heure	< 30 minutes	Outils de développement, documentation

9.2. KPIs Techniques

Performance réseau

	Objectif	Mesure actuelle	Statut
Latence AWS-cloud souverain	< 50ms (95e percentile)	15-25ms	Objectif atteint
Bande Passante	> 100 Mbps soutenu	> 150 Mbps	Objectif dépassé
Disponibilité Connectivité	99.90%	99.95%	Objectif dépassé

Conformité et Sécurité

	Objectif	Mesure actuelle	Statut
Isolation Données Sensibles	100% sur nœuds souverains	100%	Objectif atteint
Violations Politiques	99.5% (43.8h downtime/an maximum)	0%	Objectif atteint
Audit Trail	99.0% (87.6h downtime/an maximum)	100.00%	Objectif atteint

9.3 KPIs Business

Optimisation des Coûts

	Usage	Coût	Economie
AWS Managed Nodes	Workloads élastiques et non-sensibles	Instances Spot (-60% vs On-Demand)	30-40% vs infrastructure traditionnelle
Noeuds hybrides cloud souverain	Workloads sensibles et critiques	Tarification EKS Hybrid Nodes dégressive	Souveraineté garantie + performance stable

	Infrastructure	Opération	Conformité
Économies Globales	30-40% vs full on-premises	50% vs gestion multi-cloud séparée	Évite les amendes réglementaires

Évaluation des risques résiduels

Chapitre 10

SOFTEAM
UNE FILIALE DE DOCAPOSTE

10. Évaluation des risques résiduels

Malgré les mécanismes de résilience et les contrôles de sécurité mis en place, toute architecture hybride comporte des risques résiduels qu'il est essentiel d'identifier et d'évaluer. Cette section présente un début analyse des risques techniques et business qui subsistent après l'implémentation des mitigations décrites dans ce document. L'objectif n'est pas d'éliminer tous les risques (ce qui serait irréaliste) mais de les rendre acceptables et maîtrisables par rapport aux bénéfices apportés. Chaque risque est évalué selon sa probabilité d'occurrence, son impact potentiel, et les mesures de mitigation en place, permettant une prise de décision éclairée sur l'acceptabilité du niveau de risque résiduel.

10.1. Risques techniques acceptables

Risque	Probabilité	Impact	Mitigation	Acceptabilité
Latence réseau élevée	Faible	Moyen	Optimisation placement workloads	Acceptable
Complexité opérationnelle	Moyenne	Moyen	Automatisation + formation équipes	Acceptable
Dépendance AWS partielle	Élevée	Faible	Souveraineté données sur le cloud souverain	Acceptable
Évolution technologique	Moyenne	Faible	Veille technologique + roadmap	Acceptable

10.2. Risques business maîtrisés

Coûts
Le modèle de tarification dégressif AWS EKS Hybrid Nodes devient avantageux à l'échelle

Compétences
La formation des équipes est planifiée et budgétée dans la roadmap

Vendor lock-in:
L'architecture hybride réduit la dépendance exclusive à AWS

Évolutivité
L'architecture supporte la croissance et l'évolution des besoins

Recommandations stratégiques

Chapitre 11

SOFTEAM
UNE FILIALE DE DOCAPOSTE

11. Recommandations stratégiques

Au-delà des aspects techniques détaillés dans ce document, le succès d'une architecture hybride EKS-Cloud Souverain repose sur des facteurs organisationnels et stratégiques déterminants. Cette section synthétise les recommandations issues de notre retour d'expérience pour maximiser les chances de réussite. Ces recommandations couvrent à la fois les facteurs clés de succès à mettre en place dès le démarrage, et les actions immédiates à entreprendre pour valider la faisabilité et préparer le terrain. L'expérience montre que les projets qui négligent ces aspects organisationnels rencontrent des difficultés majeures, même avec une architecture technique solide.

11.1. Facteurs clés de succès

Gouvernance et organisation

- **Sponsor exécutif** : support de la direction pour les changements organisationnels
- **Équipe dédiée** : ressources allouées spécifiquement au projet hybride
- **Formation continue** : montée en compétences sur les technologies hybrides
- **Change management** : accompagnement des équipes dans la transformation

Approche technique

- **Start small** : commencer par un POC limité avant l'industrialisation
- **Automatisation :** privilégier les contrôles automatisés aux processus manuels
- **Monitoring proactif** : surveiller les métriques de performance et conformité
- **Documentation** : maintenir une documentation technique à jour

11.2. Recommandations immédiates

1 - Validation POC prioritaire (4-6 semaines)

Objectif
Valider la faisabilité technique sur un environnement de test isolé avant tout investissement majeur.

Actions concrètes

- Déployer un cluster EKS de test avec 2 nœuds hybrides
- Valider la connectivité Direct Connect et le failover VPN
- Tester les scénarios de déploiement d'applications sensibles
- Mesurer les performances réseau en conditions réelles

Critères de succès

- Latence réseau < 50ms en continu sur 48h
- Basculement réseau automatique < 30 secondes
- Déploiement d'application sensible bloqué sur nœuds AWS
- Logs sensibles routés uniquement vers le cloud souverain

2 - Formation des équipes (6-8 semaines)

Programme de formation structuré

Formations techniques requises	Durée	Public	Contenu
EKS Hybrid Nodes	3 jours	Équipes infrastructure et DevOps	Installation, configuration, troubleshooting
Cilium CNI BGP	2 jours	Équipes réseau et sécurité	Politiques réseau, BGP, observabilité
Monitoring CloudWatch	2 jours	Équipes opérationnelles	Métriques, alertes, dashboards
Sécurité Souveraineté	1 jour	Toutes les équipes	Principes, contrôles, procédures

3 - Audit sécurité complet (2-3 semaines)

Périmètre d'audit

- Évaluation des flux réseau et des politiques de sécurité
- Validation des contrôles d'accès et des mécanismes d'isolation
- Test de pénétration sur l'architecture hybride
- Revue des procédures de gestion des incidents

Conclusion et message clé

Chapitre 12

SOFTEAM
UNE FILIALE DE DOCAPOSTE

12. Conclusion et message clé

12.1. Synthèse de faisabilité technique

Au terme de cette analyse approfondie, nous pouvons affirmer que l'architecture hybride AWS EKS - cloud souverain constitue une solution techniquement viable et stratégiquement pertinente pour concilier les exigences de performance cloud et de souveraineté numérique.

Notre analyse démontre de manière factuelle que l'architecture hybride proposée répond aux trois questions fondamentales posées en introduction

Faisabilité technique démontrée

Connectivité hybride validée
L'architecture de connectivité multi-niveaux (Direct Connect + VPN redondant) permet d'atteindre les performances requises avec un SLA de 99.9%. Les tests de validation confirment une latence inférieure à 50ms et une bande passante supérieure à 100Mbps, respectant les exigences AWS EKS Hybrid Nodes.
Gestion d'identité sécurisée
AWS Systems Manager Hybrid Activation offre une solution d'authentification robuste sans stockage de credentials. La rotation automatique des tokens et l'intégration native avec EKS Hybrid Nodes éliminent les risques de sécurité traditionnels des architectures hybrides.
Isolation des données garantie
Les mécanismes de contrôle granulaires (OPA Gatekeeper, Cilium Network Policies, routage sélectif des logs) assurent une isolation stricte des données sensibles. Les tests de validation confirment qu'aucune donnée de niveau souverain ne peut transiter par des nœuds AWS.

Mise en oeuvre structurée

Roadmap pragmatique validée
La roadmap de déploiement en 4 phases sur 18-22 semaines a été conçue pour minimiser les risques.
Chaque phase dispose de critères de validation clairs et de procédures de rollback documentées.
Outils et technologies éprouvés
L'architecture s'appuie exclusivement sur des technologies matures et supportées : Ubuntu 24.04 LTS, Cilium CNI, OPA Gatekeeper, et les services AWS managés. Cette approche conservative minimise les risques technologiques.

Points de vigilance maitrisés

SPOF identifiés et couverts
L'analyse exhaustive a identifié 6 points de défaillance unique avec leurs contournements spécifiques. Les SPOF critiques (connectivité réseau, plan de contrôle EKS, endpoints SSM) disposent de mécanismes de résilience automatiques.
Retour d'expérience intégré
Les points de vigilance critiques (DNS failover SSM, placement de pods complexe, impact CPU WireGuard, gestion BGP failback) sont documentés avec leurs solutions éprouvées.

12.2. Valeur ajoutée stratégique

Avantages concurrentiels identifiés

Différenciation technologique
Cette architecture positionne l'organisation comme précurseur dans le cloud souverain, ouvrant de nouvelles opportunités business tout en respectant les contraintes réglementaires.

Flexibilité opérationnelle
La granularité des contrôles par application et par image de conteneur offre une flexibilité unique pour gérer différents niveaux de sensibilité des données.

Évolutivité future
L'architecture est compatible avec les futures exigences de conformité et peut s'étendre vers plusieurs clouds souverains (NumSpot, OVHcloud, Scaleway) ou des environnements edge ou on-prem.

Innovation préservée
Les équipes continuent de bénéficier de l'écosystème AWS (services managés, outils de développement, marketplace) tout en respectant la souveraineté.

12.3. Message clé et conclusion finale

L'architecture hybride EKS-cloud souverain est techniquement réalisable et stratégiquement pertinente. Elle résout le paradoxe apparent entre innovation cloud et souveraineté numérique, offrant une voie pragmatique pour les organisations soumises à des exigences de conformité élevées.

Il n'est pas nécessaire de choisir entre performance et souveraineté - les deux peuvent coexister dans une architecture hybride bien conçue, avec des contrôles appropriés et une vigilance opérationnelle.

Cette architecture ouvre la voie à une nouvelle génération de solutions cloud qui respectent les exigences de souveraineté tout en préservant les avantages de l'innovation technologique. Elle démontre que la conformité réglementaire peut être un facteur de différenciation plutôt qu'une contrainte limitante.

Principe directeur confirmé
La technologie reste un levier au service des objectifs métier. L'architecture proposée privilégie l'efficacité, la sobriété et l'alignement sur les besoins réels, plutôt que la complexité technique pour elle-même.

Expertise Cloud chez SOFTEAM

SOFTEAM
UNE FILIALE DE DOCAPOSTE

Expertise Cloud Softeam

Les experts Cloud Softeam accompagnent les organisations dans leur transformation digitale avec une expertise reconnue en architecture cloud, DevOps et sécurité. Notre équipe d'experts certifiés AWS, Kubernetes et sécurité vous aide à concevoir et déployer des architectures innovantes qui respectent vos contraintes métier et réglementaires.

Nos Services Cloud

Architecture Cloud

Conception d'architectures hybrides et multi-cloud

Sécurité & Conformité

Mise en conformité RGPD, HDS, ISO 27001

DevOps & Automation

Industrialisation des déploiements et opérations

Monitoring & Observabilité

Supervision et optimisation des performances

Notre expertise EKS hybrid nodes

- Implémentation réelle validée
- Retours d'expérience documentés et partagés
- Formation et accompagnement des équipes
- Support technique spécialisé

Présentation Softeam

Softeam est la filiale du conseil et services de Docaposte (Groupe La Poste).

Nous accompagnons les acteurs publics et privés dans la **mise en œuvre de leurs projets de transformation numérique**. En mobilisant des expertises sectorielles et technologiques, nous proposons une offre de conseil et de services créatrice de valeur, pour construire ensemble un **numérique responsable, durable et au service de l'intérêt général.**

Concrètement, cela se traduit dans notre attitude et dans nos engagements. Nous sommes **des partenaires de confiance** pour nos clients, en adoptant une posture d'écoute active et en coconstruisant des solutions qui répondent à leurs besoins spécifiques.

Nous sommes conscients que les défis auxquels nous faisons face sont complexes et évolutifs :
la force de notre collectif de consultants experts et engagés pour le bien commun est le socle de notre mission.

Ensemble, nous voulons faire du numérique un levier pour demain. Et cela commence justement par un bon accompagnement incluant le Cloud.

Ce guide technique est basé sur une implémentation réelle et des retours d'expérience concrets. Les métriques de performance et les configurations techniques présentées ont été validées en environnement de production. L'architecture décrite constitue un modèle de référence qui doit être adapté selon vos contraintes spécifiques.

SOFTEAM
UNE FILIALE DE DOCAPOSTE

Vous souhaitez échanger avec nos experts sur vos besoins d'accompagnement Cloud ?

Nos experts prennent (vraiment) le temps de vous écouter et comprendre vos enjeux, votre environnement et vos objectifs.

Contactez nous directement via :
ld-cloud@softeam.fr

Retrouvez toutes les actus Cloud sur notre site ou notre page LinkedIn

https://www.softeam.com/cloud

/company/softeam