



## Veille Technologique Quotidienne

### Technique

30 septembre 2026 – Fréquence : Quotidienne (du Lundi au Vendredi)

#### Synthèse & Sujet du Jour

**Sécurité de la Software Supply Chain sur OpenShift : Tekton Chains, Sigstore (RHTAS) et SBOM/VEX**

Provenance SLSA signée par Tekton Chains, signature keyless Sigstore (Red Hat Trusted Artifact Signer), SBOM/VEX avec Trusted Profile Analyzer et politique de release Enterprise Contract (Conforma) : la chaîne de build OpenShift comme frontière de confiance.

## 1. Périmètre

**Famille de veille** : 2. Métiers, Compétences & Technologies.

**Fréquence d'exécution** : Quotidienne (du Lundi au Vendredi à 08h30).

**Diffusion** : [Radar Web Aperta](#) — [LinkedIn Aperta Scientia](#) — [X @ApertaScientia](#).

## 2. Sujet du Jour — Article Publié sur le Radar Web

**Sécurité de la Software Supply Chain sur OpenShift : Tekton Chains, Sigstore (RHTAS) et SBOM/VEX**

SolarWinds (2020), Log4Shell (CVE-2021-44228), la porte dérobée XZ Utils (CVE-2024-3094) ou les campagnes de typosquatting sur npm et PyPI convergent vers un même constat d'ingénierie : **l'artefact exécuté en production n'est plus le code que l'on croit avoir écrit**. La question n'est plus "mon application est-elle vulnérable ?" mais "puis-je prouver d'où vient ce binaire, qui l'a construit, avec quelles dépendances, et qu'il est resté intact entre le commit et le Pod ?". C'est précisément le rôle d'une chaîne d'approvisionnement logicielle outillée. Sur OpenShift, l'assemblage est désormais mature : **OpenShift Pipelines (Tekton)** pour l'exécution déclarative, **Tekton Chains** pour la provenance SLSA signée, **Sigstore / Red Hat Trusted Artifact Signer (RHTAS)** pour la signature keyless, **Red Hat Trusted Profile Analyzer (TPA)** pour les SBOM et VEX, et **Enterprise Contract (désormais Conforma)** comme porte de release.

### Le pipeline comme frontière de confiance : Pipelines-as-Code

La première rupture avec les pratiques historiques consiste à retirer toute construction d'image des postes de travail :

- **Exécution conteneurisée par construction** : Une Task Tekton s'exécute dans un Pod dédié, avec son ServiceAccount, ses Workspaces et ses limites de ressources. Il n'existe plus de `docker build` local non traçable ni de tag `latest` flottant sur un laptop.
- **Pipelines-as-Code (PaC)** : les définitions de PipelineRun vivent dans le dépôt applicatif, sous `.tekton/`, et sont déclenchées par des webhooks Git (push, pull request) via la CRD `Repository` et un filtrage CEL sur les

événements. Le pipeline devient un artefact revu en *pull request*, au même titre que le code.

- **Traçabilité native** : chaque exécution produit un `TaskRun`/`PipelineRun` immuable, horodaté, référençant l'image exacte du builder (buildah, git-clone), les paramètres et le commit SHA. C'est la matière première de la provenance SLSA.

Point d'attention terrain : sans politique de *pinning* des `taskRef` (bundles OCI versionnés, digests plutôt que tags), un pipeline "sécurisé" peut exécuter une Task modifiée en amont. C'est un vecteur d'attaque direct sur la chaîne de build.

## Provenance SLSA signée par Tekton Chains

**Tekton Chains** est un contrôleur qui observe les `TaskRun` et `PipelineRun` du cluster et transforme leurs résultats en attestations de provenance **in-toto**, au format **SLSA v1.0** :

- **Attestation automatique** : Chains capture les entrées (images de base, sources, paramètres) et les sorties (digests d'images, artefacts) puis produit un document de provenance signé, sans intervention du développeur.
- **Backends de signature** : x509 *keyless* (OIDC → Fulcio, certificat à courte durée de vie), KMS managé (AWS KMS, GCP KMS, HashiCorp Vault) ou clé Kubernetes. Ce choix conditionne le niveau SLSA atteignable.
- **Publication OCI** : par défaut, l'attestation et sa signature sont poussées dans le registre sous forme d'artefacts OCI au format attendu par `cosign`, en parallèle de l'entrée correspondante dans le journal de transparence. L'image porte alors l'annotation `chains.tekton.dev/signed: "true"`.
- **Lecture SLSA** : L1 = provenance générée ; L2 = build hébergé + provenance signée ; L3 = build isolé, provenance non falsifiable. Tekton Chains sur OpenShift fournit les briques L2/L3, à condition de durcir l'isolation des builders.

## Signature keyless : Sigstore et Red Hat Trusted Artifact Signer

Le modèle Sigstore résout le problème historique de gestion des clés de signature (émission, rotation, révocation) :

- **Fulcio** émet un certificat x509 éphémère à durée de vie très courte, lié à l'identité OIDC de l'opérateur ou du workload.
- **Rekor** enregistre chaque signature dans un journal de transparence append-only à arbre de Merkle : la preuve d'existence est publique et vérifiable, la compromission devient détectable.
- **RHTAS** est la distribution entreprise supportée par Red Hat de Sigstore (Fulcio, Rekor, Trillian, TUF pour la distribution de la racine de confiance, `cosign`, `gitsign`, et le binaire `ec`). `gitsign` permet de signer les commits Git par simple authentification OIDC, sans clé locale.

Vérification en ligne de commande :

```
cosign verify ghcr.io/org/app@sha256:<digest> \
  --certificate-identity-regexp '^https://github.com/org/.*' \
  --certificate-oidc-issuer https://token.actions.githubusercontent.com
```

**Application à l'admission** : le *Sigstore Policy Controller* (CRD `ClusterImagePolicy`) refuse tout déploiement dont l'image n'est pas signée par une identité approuvée et journalisée dans Rekor. RHACS propose également la vérification de signature Cosign dans son contrôleur d'admission. La conformité n'est plus un rapport, c'est un **deny**.

## SBOM et VEX : Red Hat Trusted Profile Analyzer

Un SBOM n'est utile que s'il est exploitable :



- **Formats** : SPDX 2.3 et CycloneDX 1.6 couvrent désormais la quasi-totalité des outils de génération ; le contenu doit être attaché à l'image (attestation) plutôt que stocké hors chaîne.
- **TPA (Trustify)** centralise les SBOM produits par vos builds *et* ceux fournis par les éditeurs, puis les corrèle en continu avec les bases de vulnérabilités et les avis de sécurité (dont ceux de Red Hat).
- **VEX (Vulnerability Exploitability eXchange)** est la pièce manquante : un composant peut être présent dans l'image sans être atteignable (`vulnerable_code_not_in_execute_path`) ni même compilé (`vulnerable_code_not_present`). Sans VEX, une équipe noie ses développeurs sous des CVE non exploitables ; avec VEX, elle hiérarchise ce qui est réellement dans la surface d'attaque et répond factuellement à "*quel est le rayon d'impact d'une nouvelle CVE ?*".
- **Interrogation par PURL** : la corrélation par `pkg:npm/...`, `pkg:maven/log4j-core@2.14.1` permet de lister en quelques secondes toutes les images concernées par une alerte critique, y compris dans les registres distants.

## Enterprise Contract (Conforma) : la politique comme porte de release

La vérification doit être bloquante et identique partout :

- **ec validate image / ec verify** s'exécute dans le pipeline avant toute promotion, et refuse la release si une règle échoue.
- **Règles typiques** : signature vérifiée, provenance SLSA d'un niveau minimal, SBOM présent et attaché, absence de CVE de sévérité critique, et liste blanche des Task bundles Tekton autorisés.
- **Politiques Rego/OPA packagées en bundles OCI** : versionnées, revues, diffusable de manière incrémentale — du poste de développement (shift-left) jusqu'à l'admission cluster et au registre. Les mêmes règles peuvent être réutilisées par Quay (immuabilité, scan Clair, signature) pour verrouiller le registre.

## Chaîne complète de référence

1. Commit signé (gitsign, identité OIDC).
2. Déclenchement PaC sur webhook Git → PipelineRun versionné.
3. Build d'image (Buildah, rootless) + scan de vulnérabilités.
4. Génération SBOM et attestation.
5. Tekton Chains → provenance SLSA signée (Fulcio) + entrée Rekor + poussée OCI vers Quay.
6. **ec validate image** : porte de release bloquante.
7. Admission ClusterImagePolicy : refus de tout artefact non signé ou non conforme.
8. Corrélation SBOM/CVE/VEX continue dans TPA.

## 3. Impact sur les Coursus Aperta Scientia

### Coursus AS200 — DevOps Platform Engineer

Cette veille alimente directement les modules d'ingénierie de plateforme et de sécurisation avancée de la chaîne de livraison logicielle :

- **Construction & CI/CD (D0288)** : Conception de pipelines déclaratifs Tekton, gestion des builds conteneurisés en environnement sans privilèges (*rootless*) et automatisation via Pipelines-as-Code.



- **Exploitation & Durcissement (D0380)** : Gestion multi-cluster, observabilité de la chaîne de livraison et politiques de conformité à l'échelle.
- **Sécurisation Conteneurs & Plateforme (D0430)** : Implémentation du contrôle d'admission (Sigstore Policy Controller, RHACS), signature keyless d'artefacts (RHTAS), génération de SBOM et analyse continue d'exploitabilité (TPA / VEX).
- **Certifications préparées** : Préparation opérationnelle aux examens **EX188** (Containers), **EX280** (OpenShift Administration) et **EX288** (OpenShift Development).

## Cursus AS300 — AI Platform Engineer

La démarche de sécurisation de la supply chain logicielle est transposée à l'infrastructure d'inférence et aux pipelines MLOps :

- **Traçabilité des dépendances GPU & ML** : Audit de provenance et validation SLSA des images de base contenant les runtimes CUDA, ROCm et PyTorch.
- **Signature & Contrôle des Modèles** : Signature cryptographique des poids de modèles et des runtimes de serving vLLM / KServe avant déploiement en cluster.
- **Gouvernance & SBOM IA** : Inventaire automatisé des licences et détection proactive des vulnérabilités sur la pile MLOps (KubeFlow Pipelines, Ray, Triton).

## 4. Sources

- Red Hat Developer — Red Hat Trusted Software Supply Chain is now available — [Lien](#)
- Red Hat Developer — Trusted Software Supply Chain — [Lien](#)
- Red Hat Developer — Trusted Artifact Signer (RHTAS) — [Lien](#)
- Red Hat Developer — Trusted Profile Analyzer (TPA) — [Lien](#)
- Tekton Chains — Documentation — [Lien](#)
- Sigstore — Documentation — [Lien](#)
- SLSA — Supply-chain Levels for Software Artifacts — [Lien](#)
- Enterprise Contract / Conforma — [Lien](#)
- CycloneDX / SPDX — [Lien](#), [Lien](#)

### Contacts & Communauté :

- Direction des Formations : [learning@apertascientia.io](mailto:learning@apertascientia.io)
- Référent Handicap & Bien-être (Care) : [care@apertascientia.io](mailto:care@apertascientia.io)
- X (Twitter) : [@ApertaScientia](https://twitter.com/ApertaScientia)
- Communauté Discord : [discord.gg/eDJzR5Vk85](https://discord.gg/eDJzR5Vk85)