Veille DevOps – Avril 2025 : Nouveautés et Évolutions

Avril 2025 a été particulièrement riche en annonces et évolutions dans l'écosystème DevOps. Plusieurs outils phares de CI/CD ont introduit de nouvelles fonctionnalités améliorant la sécurité et la productivité des pipelines. Du côté de la conteneurisation, Kubernetes a livré une version majeure avec des avancées notables en stabilité et en sécurité, tandis que l'écosystème Docker a évolué pour intégrer l'IA et adapter ses politiques d'utilisation. L'Infrastructure as Code a également connu des changements structurants : **HashiCorp Terraform** poursuit son développement malgré l'essor d'une alternative open source communautaire, **OpenTofu**, désormais sous l'égide de la CNCF. En parallèle, **Ansible** prépare une mise à jour majeure de son moteur, et **Pulumi** enrichit son offre avec l'IA et une meilleure interopérabilité avec Terraform. Enfin, l'accent reste mis sur l'**observabilité** unifiée et la **sécurité DevSecOps** : OpenTelemetry s'impose comme standard de fait pour les traces, métriques et logs, et plusieurs alertes et rapports soulignent l'importance de sécuriser la chaîne logicielle face à des menaces toujours plus sophistiquées.

CI/CD : évolutions des pipelines et déploiements continus

- **Jenkins** – Le célèbre serveur CI a publié en avril une mise à jour LTS axée sur la performance et la fiabilité. Jenkins 2.504.1 (30 avril 2025) apporte un correctif majeur améliorant le traitement des nœuds build afin d'éviter des scans inutiles qui dégradaient les performances sur les installations à grande échelle. Cette LTS intègre aussi la montée de version de composants internes (Jetty 12, Winstone 8.7) pour renforcer la stabilité. Côté interface, l'équipe poursuit la modernisation : l'ancienne bibliothèque Yahoo UI a été retirée du core, et un widget de détails de build *expérimental* fait son apparition. Jenkins a également diffusé en avril plusieurs correctifs de sécurité pour remédier à des vulnérabilités signalées, rappelant la nécessité de maintenir les instances CI à jour.

- **GitLab CI/CD** – La version **GitLab 17.11** sortie le 17 avril 2025 apporte plus de 60 améliorations, dont de nouvelles **entrées de pipeline CI/CD (pipeline inputs)**. Cette fonctionnalité permet aux développeurs de paramétrer dynamiquement le comportement d'un pipeline via des `inputs` structurés plutôt que d'utiliser des variables pipeline classiques, évitant ainsi les risques liés à la surpriorisation de ces variables. Il devient même possible de désactiver totalement l'usage des variables pipeline au profit des *inputs* pour un contrôle plus sûr. GitLab 17.11 introduit également des *frameworks* de conformité personnalisés et des comptes de service améliorés, préfigurant les avancées attendues dans GitLab 18.0 (prévue en mai) qui mettra l'accent sur la sécurité et l'IA. À noter qu'un correctif de sécurité a été publié en avril pour GitLab (17.10.3) afin de corriger des failles, dont des XSS, confirmant l'attention portée à la **sécurité by design** dans la plateforme.

- **GitHub Actions** – La plateforme CI/CD de GitHub a évolué pour offrir davantage de **contrôle et sécurité des workflows**. Depuis mi-avril, GitHub a introduit un mécanisme obligeant à approuver

manuellement l'exécution de workflows déclenchés par des actions de GitHub Copilot (génération automatique). En pratique, tout événement initié par l'IA ne lance plus automatiquement de pipeline sans validation d'un mainteneur, évitant l'exécution aveugle de code potentiellement malveillant. Par ailleurs, GitHub a mis à jour son **Actions Runner Controller** (outil de gestion des runners auto-hébergés) afin d'améliorer les performances et la compatibilité avec les déploiements Kubernetes récents. Côté environnements, on note des changements majeurs : l'image Ubuntu 20.04 a été **retirée le 15 avril 2025** et n'est plus supportée, forçant les pipelines à migrer vers Ubuntu 22.04 ou versions ultérieures. À l'inverse, une **image Windows Server 2025** a fait son apparition dès le 8 avril, permettant d'exécuter des jobs sur cet OS de prochaine génération via le label `runs-on: windows-2025`. Ces évolutions illustrent la volonté de GitHub d'aligner ses outils CI/CD sur les nouveautés du marché tout en supprimant les stacks obsolètes pour des raisons de maintenance et de sécurité.

- **Argo CD** – L'outil de déploiement continu GitOps maintenu par la CNCF franchit un cap important. Lors de KubeCon EU 2025 (1er avril), la sortie imminente d'**Argo CD v3.0** a été annoncée. La première Release Candidate (3.0.0-rc1) est disponible pour tests, la version finale GA étant prévue début mai. Il s'agit de la **première mise à jour majeure en 4 ans** pour Argo CD, marquant un bond en termes de **scalabilité, performance et sécurité**. Argo CD v3 renforce notamment les mécanismes d'authentification et d'autorisation et réduit son empreinte mémoire pour mieux supporter les déploiements massifs. L'objectif affiché est d'aller « au-delà du GitOps » en accélérant les déploiements Kubernetes tout en réduisant les coûts opérationnels. Côté supply chain, le projet adopte une posture DevSecOps exemplaire : toutes les images conteneur officielles Argo CD sont désormais **signées via cosign** et accompagnées de métadonnées de provenance conformes SLSA niveau 3. Cela garantit l'intégrité des artefacts déployés en production. La communauté est encouragée à tester la RC et à remonter ses retours afin de fiabiliser la version stable à venir. Argo CD v3 s'annonce comme un jalon majeur pour l'écosystème GitOps, qui devra observer sa maturité (encore *release candidate*) avant une adoption large en production.

Conteneurisation et orchestration : vers plus de maturité

- **Kubernetes 1.33 "Octarine"** – La version mineure 1.33 du moteur d'orchestration de conteneurs est sortie le **23 avril 2025** et apporte **64 améliorations** (18 fonctionnalités stables, 20 en bêta, 24 en alpha) couvrant la sécurité, l'efficacité et l'expérience développeur. L'une des avancées les plus attendues est la promotion en **stable du support natif des conteneurs "sidecar"** dans les Pods. Désormais, on peut déclarer officiellement des conteneurs auxiliaires dont le cycle de vie est géré plus finement par Kubernetes, sans recourir à des contournements. Cela facilite des modèles comme les proxies de service mesh ou les agents de logging, en assurant qu'un sidecar démarre et s'arrête de façon coordonnée avec le conteneur principal. Côté sécurité, **les "User Namespaces" sont maintenant activés par défaut**, ce qui permet une isolation renforcée en exécutant les conteneurs avec des UID/GID mappés au niveau du noyau (rootless containers), réduisant l'impact en cas d'évasion. Kubernetes 1.33 introduit aussi en alpha la possibilité de **redimensionner un Pod à chaud** (in-place pod resizing), afin de modifier les ressources

CPU/mémoire d'un conteneur sans le recréer – une fonctionnalité très attendue pour l'auto-scalabilité verticale. Sur le plan réseau, l'abandon progressif de l'API Endpoints au profit des **EndpointSlices** s'achève : l'ancienne API est désormais marquée *deprecated* et sera retirée, les EndpointSlices offrant une meilleure échelle et la prise en charge des clusters multi-réseaux. De même, le type de volume `gitRepo` a été **supprimé** pour des raisons de sécurité (il est recommandé de le remplacer par un initContainer effectuant un `git clone`). Enfin, le **support de Windows en mode host networking a été retiré** en raison de limitations techniques sur cette plateforme. L'ensemble de ces changements témoignent d'un Kubernetes qui gagne en maturité opérationnelle (cycles de vie de conteneurs améliorés, nettoyage d'anciennes fonctionnalités risquées). Les équipes doivent toutefois vérifier la compatibilité de leurs manifestes et contrôleurs avec ces évolutions (notamment la fin des Endpoints API). Kubernetes 1.33 confirme en tout cas la tendance à une standardisation des patterns (sidecars natifs, sécurité renforcée) et à une convergence avec l'écosystème cloud-native (interopérabilité accrue avec OpenTelemetry, cf. Observabilité). Son adoption en production dépendra de la rapidité des distributions Kubernetes (clouds, OpenShift, etc.) à l'intégrer et de la stabilité constatée des features alpha/bêta sur le terrain.

- **Docker et conteneurs** – En avril, Docker Inc. a publié la version **Docker Desktop 4.41** (29 avril 2025) avec des nouveautés ciblant les cas d'usage IA et la facilitation du travail en équipe. Une fonctionnalité phare est l'arrivée de **Docker Model Runner sur Windows** : jusqu'ici réservé à Linux, cet outil permettant d'orchestrer localement des modèles de machine learning dans des conteneurs prend désormais en charge Windows, avec support de l'accélération GPU NVIDIA. Couplé à cela, Docker Desktop intègre une section "Models" dans son interface, offrant aux développeurs une vue unifiée pour gérer leurs modèles ML comme des artefacts de première classe aux côtés des conteneurs et images classiques. Il devient possible de **pusher un modèle sur Docker Hub** exactement comme on pousserait une image (`docker model push <model>`), ce qui unifie les workflows et encourage le partage de modèles IA via les mêmes pratiques que l'infrastructure conteneurisée. Par ailleurs, cette version facilite l'intégration des modèles dans les applications : on peut déclarer un service de type "AI model" directement dans un fichier **Docker Compose**, ou utiliser les bibliothèques **Testcontainers** (Java, Go, etc.) pour manipuler les modèles en test unitaire de façon similaire à une base de données éphémère. Ces ajouts soulignent l'effort de Docker pour se positionner sur le MLOps et rendre l'IA portable et testable via les outillages DevOps existants. Sur un volet plus réglementaire, Docker a annoncé une évolution de ses **limitations Docker Hub** à compter du 1er avril 2025. Désormais, les utilisateurs non authentifiés sont restreints à **10 pulls par heure** (au lieu de 100 par 6 heures auparavant), tandis que les utilisateurs gratuits connectés bénéficient d'un quota relevé (100 pulls/heure). Les abonnés payants (Pro/Team/Business) conservent un accès illimité (*fair use*). Cette nouvelle politique vise à encourager la connexion aux comptes Docker Hub pour un meilleur suivi, tout en prévenant les abus. Elle impose aux infrastructures CI/CD et aux utilisateurs heavy de mettre en cache les images ou de s'authentifier pour éviter les taux-limit. Enfin, bien que Docker Swarm n'ait pas fait l'objet d'annonces majeures (cet orchestrateur restant en maintenance avec une communauté réduite), on observe l'essor continu d'outils comme **Podman** en alternative légère à Docker Engine, et l'adoption de plus en plus native de **containerd** et **CRIo** dans les distributions – ces

tendances se sont confirmées lors des échanges communautaires d'avril, même si aucune sortie de version notable n'est à signaler ce mois-ci.

Infrastructure as Code : évolutions des outils et nouvelles communautés

- **Terraform** – L'outil IaC de HashiCorp continue d'évoluer dans la lignée de la version 1.x. En avril, plusieurs versions préliminaires ont été publiées en vue de Terraform **1.12**. La release candidate 1.12.0-rc1 (30 avril 2025) apporte par exemple un nouveau **backend de stockage distant pour Oracle Cloud (OCI Object Storage)**, élargissant le choix d'emplacements où stocker l'état Terraform. Parallèlement, les développeurs Terraform introduisent des améliorations du côté tests et langage : la commande `terraform test` supporte maintenant un paramètre de **parallélisme** pour exécuter les plans en test plus rapidement sur plusieurs threads. Le langage HCL voit l'optimisation de ses opérateurs booléens avec une évaluation *lazy* (court-circuit logique) qui évite des calculs inutiles dans les expressions conditionnelles. Autre ajout notable : les blocs `import` (permettant d'importer un état existant dans la configuration) supportent désormais un attribut *identity* en plus de l'ID, pour une identification plus souple de la ressource cible. Ces ajouts mineurs améliorent l'ergonomie et la performance de Terraform, qui poursuit donc son chemin de maturité fonctionnelle. Toutefois, l'actualité Terraform ne se limite pas au produit en lui-même – la communauté a été marquée par les suites du changement de licence de HashiCorp (passage en source disponible, licence BSL, en 2023). En réaction, le projet **OpenTofu** (fork open source de Terraform) a gagné en **traction**. En avril 2025, **OpenTofu a officiellement rejoint la CNCF en tant que projet Sandbox**, signalant l'engagement de la communauté cloud-native à soutenir une alternative ouverte et neutre à Terraform. Pendant KubeCon EU, un *OpenTofu Day* a même été organisé le 1er avril à Londres, preuve de l'intérêt croissant : de grands acteurs commencent à l'adopter (par exemple, la presse spécialisée rapporte que Fidelity Investments migre de Terraform vers OpenTofu pour ses besoins IaC). L'entrée d'OpenTofu à la CNCF devrait accélérer son écosystème (gouvernance ouverte, contributions) mais il reste encore en phase de stabilisation (version 1.10 alpha en test au 25 avril). On peut s'attendre à une coexistence des deux outils dans les mois à venir : Terraform conserve une large base installée et des sorties régulières, tandis qu'OpenTofu apporte une assurance pérenne aux entreprises voulant éviter la dépendance à HashiCorp. Les roadmaps des deux projets (Terraform 1.12+ et OpenTofu 1.x) seront à surveiller pour voir s'ils divergent ou restent compatibles sur le plan technique.

- **Ansible** – Le monde de la configuration déclarative a aussi son lot de nouveautés. **Ansible Core 2.19** est en préparation active au sein de la communauté Red Hat/Ansible. En effet, la branche de développement a atteint la phase de *feature freeze* mi-avril 2025, avec une bêta disponible pour tests. Cette version 2.19 de l'exécuteur Ansible embarque une refonte majeure baptisée **Data Tagging**. Selon les développeurs, il s'agit de la plus grande transformation du cœur Ansible depuis l'introduction des *collections* en 2020. Concrètement, Data Tagging apporte un **remaniement complet du moteur de templating Jinja2 et du traitement des variables**, afin d'attacher des métadonnées de type aux données manipulées. L'objectif est d'éviter les conversions implicites et ambigüités (tout traiter en texte) en conservant la nature des données à

travers les templates – ce qui améliore la fiabilité des playbooks et la sécurité (moins de surprises d'exécution). Cette évolution de fond a des impacts en chaîne : certaines collections Ansible ont dû s'adapter car des comportements changent légèrement dans la manière dont les variables et contextes sont gérés. La communauté a été invitée à tester la pré-version (`ansible-core 2.19.0b1`) pour repérer d'éventuelles incompatibilités dans les playbooks existants. En parallèle du moteur, l'**Ansible Community Package 12** (le paquet Ansible complet avec collections) est d'ores et déjà proposé en pré-release 12.0.0a1, incluant ce core 2.19 bêta. On anticipe qu'Ansible 2.19 final et le package Ansible 12 sortiront peu après, possiblement en mai/juin, une fois la stabilisation achevée. Ces versions marqueront un pas en avant en termes de maintenabilité et robustesse des automations. Du côté de l'**Automation Platform** de Red Hat (offre entreprise d'Ansible), la version 2.5 continue d'évoluer avec des correctifs mensuels – aucune annonce majeure en avril, si ce n'est la confirmation que la **version open source Ansible 2.18.x (community 11.x)** reste supportée un peu plus longtemps que prévu (l'EOL de Ansible 2.18 a été repoussé à fin avril 2025 pour laisser le temps de migrer vers 2.19). En somme, Ansible franchit un cap technique important, à suivre de près pour adapter les playbooks et plugins, mais témoignant d'une bonne vitalité du projet open source soutenu par Red Hat et sa communauté.

- **Pulumi** – La plateforme IaC "as Code" multi-langages Pulumi a multiplié les annonces en avril 2025, démontrant son effort d'innovation. Pulumi a dévoilé une intégration poussée de l'**IA générative dans les workflows IaC** grâce à son **Pulumi AI / Model Context Protocol (MCP) Server**. Concrètement, Pulumi peut désormais se connecter à des assistants de code pilotés par IA (tels que GitHub Copilot, Claude, Cursor, etc.) pour fournir en temps réel des informations sur les ressources cloud et la syntaxe Pulumi directement dans l'éditeur. Cela signifie par exemple qu'en écrivant du code d'infrastructure, l'IA peut suggérer les bonnes propriétés pour une ressource ou alerter sur des paramètres manquants, en s'appuyant sur la documentation Pulumi. L'objectif est de réduire les allers-retours entre l'IDE, la doc et la console en offrant un **assistant contextuel à la IaC**, accélérant le développement d'infrastructure tout en diminuant les erreurs. En parallèle, Pulumi a introduit sa **nouvelle génération de Composants** réutilisables, désormais **multi-langages par construction**. Un même composant d'infrastructure (par ex. un module provisionnant un VPC complet) peut être écrit une fois et exposé à toutes les languages Pulumi (Python, TypeScript, Go, .NET, Java, etc.) avec un packaging unifié. Ces *Component Packages* de nouvelle génération embarquent en outre de la validation d'inputs, des messages d'erreur améliorés et une documentation automatique, ce qui permet aux équipes plate-forme de distribuer des abstractions haut niveau plus facilement aux développeurs tout en assurant les bonnes pratiques (Infrastructure as Code modulaire et robuste). Pulumi continue aussi d'élargir la couverture des services cloud supportés : **27 nouveaux providers natifs** ont été ajoutés au **Pulumi Registry** en ce début de trimestre, portant le total à plus de 150 providers couvrant ~7500 types de ressources. La vague de nouveautés inclut des providers pour Jira, GitLab, Railway, PlanetScale, etc., reflétant une extension vers des services SaaS et bases de données cloud en plus des grands cloud providers. Chaque provider Pulumi étant versionné indépendamment, les utilisateurs peuvent tirer parti de ces services additionnels sans attendre une mise à jour globale de Pulumi. Enfin, Pulumi s'attaque frontalement au problème de la migration depuis Terraform avec un **convertisseur Terraform -> Pulumi grandement amélioré**. Depuis la CLI Pulumi 3.153+, il est possible d'importer automatiquement n'importe quel projet

Terraform existant *y compris* s'il utilise des providers Terraform non supportés nativement par Pulumi. Techniquement, Pulumi est capable de réutiliser **tout provider Terraform (ou OpenTofu)** via le mécanisme Bridged Provider, éliminant les limites précédentes qui forçaient à attendre un équivalent Pulumi. Ainsi, les équipes peuvent migrer leurs configurations sans devoir renoncer aux providers exotiques ou propriétaires qu'elles utilisaient sur Terraform. Combiné au support des HCL dans les convertisseurs, Pulumi tente de réduire au minimum les obstacles à l'adoption pour les organisations déjà investies dans Terraform. En somme, Pulumi consolide sa position de challenger innovant en IaC : intégration de l'IA (ce qui demeure naissant dans d'autres outils), richesse de l'écosystème et ponts vers la base installée Terraform. La maturité de ces nouvelles capacités (MCP Server, convertisseur universel) devra être éprouvée, mais elles pourraient influencer les pratiques IaC à moyen terme, en rendant l'infrastructure as code plus accessible (grâce à l'IA) et plus universelle.

- **Chef, Salt et autres** – Les outils de configuration historiques continuent leur chemin avec moins de fanfare. **Progress Chef** (suite d'automatisation après l'acquisition de Chef) n'a pas annoncé de nouvelle version produit en avril, mais a communiqué sur son programme communautaire et de support. L'entreprise a allongé la durée de vie de certaines versions pour faciliter la transition de ses clients : par exemple, la date d'end-of-life du **Chef Infra Client 17** a été repoussée au 31 mars 2025, ce qui laisse supposer que Chef Infra Client 18 (version actuelle) reste le focus jusqu'à ce qu'une v19 soit prête. Progress met en avant ses **Chef Champions 2025** (évangelistes de la communauté) pour encourager le partage de bonnes pratiques, signe que l'adoption de Chef se fait surtout via sa base installée stable. Du côté de **SaltStack** (maintenant Salt Project sous VMware), aucune annonce spécifique d'avril – le projet suit son cycle de releases trimestrielles (la 3006 sortie en février 2025). En revanche on observe un rapprochement des concepts entre ces outils "Configuration Management" classiques et l'approche Infrastructure as Code plus récente : par exemple, la notion de *state* et de *desired configuration* de Salt/Chef converge avec les pratiques GitOps/Kubernetes chez certains utilisateurs avancés. Enfin, **Crossplane** (projet CNCF permettant de piloter l'infrastructure cloud via Kubernetes) n'a pas sorti de version majeure en avril, mais continue de gagner en maturité en incubant dans la CNCF. Son modèle déclaratif par Custom Resources s'inscrit dans la tendance de "Platform Engineering" où Kubernetes devient le point central de contrôle de l'infra (on a pu le voir dans quelques présentations durant KubeCon). À défaut de nouvelles versions, il est clair que l'écosystème IaC s'enrichit : entre les solutions historiques (Chef/Puppet) stables mais discrètes, les géants comme Terraform en évolution continue, et les approches émergentes (Pulumi, Crossplane, CDK, etc.), les équipes ont un éventail d'options pour automatiser leur infrastructure. Le choix dépendra de la priorité donnée à la compatibilité open source, à la polyvalence multi-cloud ou à l'intégration avec un orchestrateur existant.

Observabilité et DevSecOps : vers une unification et une sécurisation accrues

- **Observabilité Cloud-Native (OpenTelemetry)** – Un thème marquant d'avril 2025 est la consécration d'**OpenTelemetry (OTel)** comme standard unifié pour la télémétrie. Lors de

KubeCon + CloudNativeCon Europe 2025 à Londres, il est apparu clairement qu'OpenTelemetry est désormais le **pilier central de l'observabilité cloud-native**. De nombreux projets ont annoncé s'aligner nativement sur OTel pour collecter et exporter leurs métriques, traces et logs. Par exemple, **Jaeger v2**, le système de tracing distribué, a adopté OpenTelemetry comme fondation : il peut ingérer le protocole OTLP en natif et délègue une partie du traitement (batch, transformation) à l'OTel Collector, ce qui le rend essentiellement *OTel-native* sous le capot. Côté métriques, les interactions entre **Prometheus** et OpenTelemetry continuent de s'améliorer. Une session dédiée à KubeCon a souligné les différences de paradigme (Prometheus en *pull*, OTel en *push*) et les efforts de la SIG d'interopérabilité pour les résoudre. Parmi ces efforts : le support des caractères UTF-8 dans Prometheus pour coller aux conventions OTel, et des améliorations des *exporters* OpenTelemetry vers Prometheus. L'objectif à terme est d'avoir un **échange fluide entre Prometheus et OTel**, permettant par exemple à des applications instrumentées en OTel d'alimenter des bases de données de séries temporelles Prometheus, ou inversement de convertir aisément des *metrics* Prom en OTLP. De plus, OpenTelemetry étend sa portée aux **logs** : un article de blog officiel du 18 avril *« OpenTelemetry Logging and You »* discute de la conception de l'API de Logs OTel et de son intégration avec les évènements, signe que la composante *logging* arrive à maturité. Même si les logs via OTel en sont encore au stade de standardisation (stade bêta dans certains SDKs), l'idée est d'offrir enfin un **cadre unifié pour les 3 piliers observabilité** (traces, métriques, logs) afin de corréler plus facilement ces données. À noter aussi l'essor des techniques d'observabilité **eBPF** mises en avant lors de KubeCon : des projets comme **Pixie** et **Inspektor Gadget** utilisent eBPF pour instrumenter automatiquement les applications sans code, et Pixie est en cours de refonte pour agir comme un agent universel exportant vers FluentBit, facilitant l'intégration dans les pipelines OpenTelemetry. En résumé, l'écosystème observabilité converge : OTel s'impose comme couche d'instrumentation standard, avec l'appui des grands acteurs (CNCF, éditeurs APM, etc.), tandis que des solutions techniques (eBPF, collectors unifiés) viennent enrichir les capacités de collecte. Cela bénéficie aux équipes DevOps/SRE qui gagnent un langage commun pour instrumenter et monitorer les systèmes hétérogènes. On peut s'attendre à ce que de plus en plus de solutions *propriétaires* adoptent OTel en entrée/sortie, réduisant le verrouillage et simplifiant les architectures de monitoring sur le moyen terme.

- **Métriques, dashboards et alerting** – Du côté des outils d'observabilité établis, avril 2025 n'a pas vu de version majeure de **Prometheus** ou **Grafana**, mais des évolutions incrémentales. Prometheus a reçu des mises à jour mineures (correctifs de la branche 2.32.x et 2.33) et travaille à son adaptation aux changements Kubernetes (ex : support des EndpointSlices natifs pour la découverte). **Grafana** de son côté consolide sa version 10 sortie l'an passé : de nouveaux plugins officiels sont apparus sur la marketplace en avril, et Grafana Labs a teasé quelques futures fonctionnalités (peut-être pour Grafana 11) autour de l'expérience utilisateur et de l'AIOps. Par exemple, la possibilité d'embarquer des résumés issus de modèles d'IA dans les tableaux de bord ou de détecter automatiquement des anomalies via machine learning a été évoquée lors de meetups. En attendant, l'écosystème Grafana s'enrichit via ses projets satellites : **Loki** (logs agrégés façon PromQL) a sorti une v2.9 avec des optimisations de requêtage, **Tempo** (tracing) poursuit son intégration OTel, et **Mimir** (métriques scalable) est maintenant utilisé en production par plusieurs grands comptes, montrant la viabilité des solutions OSS face aux services managés.

Ces tendances confirment une approche modulable de l'observabilité : des briques spécialisées reliées par OpenTelemetry et par Grafana en visualisation unifiée. Côté cloud, les fournisseurs comme AWS, GCP, Azure continuent d'annoncer l'adoption d'OTel pour leur instrumentation interne – par exemple, Google Cloud a indiqué lors de son newsletter d'avril que les clients de Cloud Logging pourront exporter les logs au format OTel sans surcharge, et qu'une gestion plus fine des quotas de logs par région sera introduite. Globalement, la **bonne pratique émergente** est de *penser observabilité dès la conception*, en intégrant des SDK OTel ou des sidecars d'inspection (eBPF) dès les phases de dev/test, pour éviter les angles morts en production. Les équipes DevOps renforcent ainsi leur posture *monitoring as code*, en versionnant aussi leurs configurations de dashboards, d'alertes et de SLO (par exemple via des outils comme Grafana OnCall ou des CRDs Kubernetes pour les alertes). Cette homogénéisation, bien visible dans les discussions communautaires d'avril, devrait améliorer la fiabilité des systèmes : les incidents pourront être détectés et diagnostiqués plus rapidement grâce à des traces corrélées aux métriques et logs, le tout instrumenté de façon standardisée.

- **Sécurité de la chaîne CI/CD (DevSecOps)** – Les événements d'avril 2025 ont encore mis en lumière la **menace des attaques supply chain** et l'importance d'intégrer la sécurité à chaque étape du cycle DevOps. Une alerte notable est intervenue mi-mars et a été reprise en avril : la CISA (agence cyber US) a émis un avis concernant une compromission de plugin **GitHub Actions** très utilisée (tj-actions/changed-files), marquée CVE-2025-30066. Cette attaque insidieuse injectait du code malveillant dans l'action de build pour **exfiltrer les secrets des logs** de workflow. En pratique, des clés AWS, tokens npm, PAT GitHub, etc., pouvaient être divulgués via les journaux d'exécution, compromettant potentiellement des infrastructures entières. L'incident a été ajouté au catalogue CISA des vulnérabilités exploitées activement, soulignant son sérieux. GitHub a depuis révoqué l'action affectée et communiqué sur l'importance de vérifier la provenance des GitHub Actions utilisées – c'est d'ailleurs une motivation derrière les nouvelles **protections de workflow** (cf. GitHub Actions plus haut) qui exigent des approbations manuelles pour les contributions non fiables. Dans le même registre, **Jenkins** a publié plusieurs bulletins en avril pour corriger des failles dans des plugins tiers, incitant les admins à maintenir un inventaire et à appliquer les patches de plugins CI avec diligence. Ces exemples montrent que la **sécurité des outils d'intégration** est désormais aussi critique que la sécurité du code déployé.

- **Supply Chain logicielle** – Plusieurs rapports publiés en avril tirent le signal d'alarme sur la persistance des failles supply chain. Le **Datadog State of DevSecOps 2025** (24 avril) révèle que *seuls 18%* des vulnérabilités marquées critiques dans les scanners méritent vraiment une priorité élevée, suggérant un besoin de mieux trier les alertes. Surtout, Datadog met en évidence la présence de **milliers de packages malveillants sur PyPI et npm** utilisés pour des attaques (typosquatting, dépendances piégées). Cette prolifération signifie que les pipelines CI/CD doivent intégrer des scans de dépendances et que les développeurs doivent être formés à vérifier l'authenticité des librairies open source qu'ils ajoutent. De son côté, JFrog a publié lors de KubeCon son rapport *Software Supply Chain State of the Union 2025*. Il y est question d'une **"Quad-fecta" d'exploits** menaçant la chaîne logicielle moderne, incluant la prolifération de CVE mal scorées, la difficulté à gouverner les modèles d'IA publics utilisés en dev, et les failles inédites liées à l'adoption de l'IA. En effet, de plus en plus d'entreprises incorporent des modèles pré-

entraînés (ex. GPT) ou des images Docker d'IA, parfois sans contrôle strict, créant de nouvelles surfaces d'attaque. L'appel de JFrog et d'autres experts à RSA Conference 2025 est de mettre en place des **processus outillés pour la supply chain** : signature d'images et de packages (les initiatives **Sigstore/cosign** gagnent du terrain, comme chez Argo CD), génération et vérification d'**SBOMs** (CycloneDX, SPDX), surveillance continue des dépôts de code et artefacts (par ex. GitHub Advanced Security, Snyk, etc.). En avril, **Cycode** a d'ailleurs annoncé l'ajout d'agents IA à sa plateforme de sécurité de pipeline, pour assister les devs dans la correction des faiblesses détectées. Sur le plan normatif, on note l'adoption progressive des cadres comme la **Supply Chain Levels for Software Artifacts (SLSA)** et la conformité au **NIST SSDF** – plusieurs conférences en ont traité ce mois-ci, les grandes entreprises cherchant à atteindre un niveau 2 ou 3 SLSA sur leurs projets critiques d'ici la fin de l'année.

- **DevSecOps au quotidien** – Au-delà des attaques sophistiquées, les *bonnes pratiques* DevSecOps continuent d'émerger de la communauté. En avril, beaucoup de discussions ont porté sur l'automatisation des tests de sécurité dans les pipelines : intégration systématique de tests de composition logicielle (SCA), de scans de conteneurs (Trivy, Anchore) et d'analyses statiques (Semgrep, CodeQL) à chaque build. L'accent est également mis sur la **sécurité du runtime Kubernetes** : l'admission control via **OPA/Gatekeeper** ou Kyverno pour empêcher le déploiement de pods non conformes (images non signées, privilèges trop élevés, etc.), et l'usage croissant d'**eBPF** pour la détection d'anomalies en prod (projets Cilium Tetragon, Pixie). Par ailleurs, la sensibilisation des développeurs s'intensifie : le DevSecOps met en avant la formation "shift-left", et en avril on a vu l'émergence de nouvelles plateformes de *Capture The Flag* orientées CI/CD pour entraîner les équipes à identifier et corriger des failles dans des pipelines factices. Enfin, un sujet en vogue est la protection des **données d'entraînement IA** et des **modèles** dans la chaîne DevOps – les modèles étant devenus des artefacts comme les autres, leur intégrité et leurs licences doivent être gérées (d'où des outils pour scanner les modèles d'IA à la recherche de métadonnées suspectes, ou pour vérifier les empreintes des datasets). En somme, la culture DevSecOps s'ancre davantage : on parle maintenant de *"Secure from design to deployment"*, ce qui se reflète dans la roadmap de nombreux outils (par ex. GitLab intègre des fonctionnalités de plus en plus riches de test de sécurité dans sa suite, GitHub déploie des protection par défaut, etc.). L'enjeu à moyen terme sera d'automatiser au maximum ces contrôles sans ralentir les déploiements – d'où l'adoption grandissante de l'IA pour traiter l'avalanche de résultats de scans (prioritisation intelligente des CVEs, suggestions de fix, etc.).

Communauté et événements : highlights du mois d'avril

- **KubeCon + CloudNativeCon EU 2025 (1-4 avril, Londres)** – L'événement majeur de la communauté cloud-native a rythmé la première semaine d'avril. Parmi les annonces phares déjà évoquées : la CNCF a accueilli de nouveaux projets en **Sandbox** comme OpenTofu (Terraform open source), témoignant d'une volonté d'ouverture de l'écosystème IaC. Le pré-lancement d'Argo CD 3.0 a fait sensation, tout comme les présentations soulignant la domination d'OpenTelemetry (voir Observabilité). On a également parlé **plateforme interne** (*Internal Developer Platforms*), un thème de plus en plus central : plusieurs retours d'expérience (ING,

Adidas…) ont montré comment assembler Kubernetes + CI/CD + GitOps pour offrir un **self-service aux développeurs**, ce qui s'inscrit dans la mouvance *Platform Engineering*. Au niveau CNCF, aucun projet n'a atteint le niveau *Graduated* en avril, mais **Backstage** (portail développeur) a été promu en Incubation, confirmant l'intérêt pour améliorer l'expérience DevOps. L'événement a aussi mis un coup de projecteur sur la **durabilité** et l'optimisation des coûts (FinOps) – par exemple, certains toolings comme Kubecost ou des opérateurs de scaling à la demande ont été présentés pour réduire l'empreinte des clusters. Globalement, KubeCon EU 2025 aura montré un écosystème DevOps **très mature et orienté intégration** : intégration des outils entre eux (via des standards comme OTel, des APIs uniformisées), intégration de la sécurité et de la conformité (beaucoup de talks sur la supply chain), et intégration de l'IA dans les workflows (plusieurs démos de GitHub Copilot X, d'AI assistée pour la génération de YAML Kubernetes, etc.). La communauté européenne est apparue prudente mais enthousiaste envers ces tendances, avec l'idée de tirer parti de l'IA et du cloud tout en gardant le contrôle (open source, gouvernance CNCF).

- **Autres conférences et meetups** – En dehors de KubeCon, **l'édition 2025 de RSA Conference** (conférence cybersécurité, 24-27 avril à San Francisco) a mis en avant le rapprochement entre sécurité et DevOps. Le terme *DevSecOps* y était omniprésent, avec des sessions dédiées à la sécurisation des pipelines CI/CD, la gestion des secrets et l'IA appliquée à la détection de menaces. Le **CISO de GitLab, Josh Lemos**, y a notamment évoqué l'utilisation de l'IA pour identifier plus vite les vulnérabilités dans le code et les conteneurs, ainsi que l'importance de la transparence (SBOMs) dans un monde open source. Sur le mois, de nombreux **DevOpsDays locaux** (Paris, Berlin, Austin…) ont repris en présentiel. Les thèmes récurrents dans ces meetups : l'**automatisation du déploiement sur cloud hybride**, les retours d'expérience sur l'adoption de **GitOps** à l'échelle de l'entreprise, et des discussions sur le rôle émergent de *Platform Engineer* distinct de *Site Reliability Engineer*. On constate que les organisations cherchent à industrialiser les bonnes pratiques DevOps : par exemple, des ateliers ont présenté comment définir des **Golden Paths** (parcours standardisés) pour les développeurs – c'est-à-dire fournir des modèles de pipelines CI/CD, de configurations Terraform ou Kubernetes validés par l'entreprise, afin d'accélérer les nouveaux projets sans repartir de zéro.

- **Éducation et certifications** – Avril n'a pas vu l'introduction de nouvelle certification officielle, mais plusieurs programmes ont évolué. La CNCF a annoncé une mise à jour prochaine des examens **CKA/CKAD** pour couvrir les nouveautés de Kubernetes 1.33, garantissant que les ingénieurs certifiés maîtrisent les sidecars et EndpointSlices par exemple. Du côté de HashiCorp, l'examen **Terraform Associate** reste basé sur la v1.1 (MPL) pour l'instant, en attendant peut-être une version alignée sur Terraform 1.5+ ou même OpenTofu si la demande se fait sentir. La **Linux Foundation** a lancé une formation *free online* en avril sur la sécurité des supply chains (inspirée par le programme OpenSSF), accessible à tous pour promouvoir les fondamentaux (signatures, SBOM, etc.). On note également l'essor de formations spécialisées courtes : "CI/CD Security", "Kubernetes Observability", proposées par des cabinets ou via des MOOCs, signe que la spécialisation DevOps se raffine. Enfin, certains éditeurs profitent de la vague IA : GitHub a par exemple ouvert une **beta de certification "GitHub Copilot"** pour attester de la capacité à intégrer l'IA dans son workflow de développement. Si cela reste anecdotique pour l'instant, cela préfigure possiblement de nouveaux rôles ou du moins de nouvelles compétences attendues

dans les équipes (savoir utiliser efficacement les outils dopés à l'IA, tout en respectant la gouvernance et la sécurité).

En résumé, cette veille d'avril 2025 montre un paysage DevOps en pleine consolidation et transition vers de nouvelles possibilités. Les outils matures (Kubernetes, Terraform, Ansible, Jenkins…) continuent d'évoluer sans rupture, en insistant sur la performance, l'UX et la sécurité – ce qui les rend toujours plus **fiables et "enterprise-ready"**. Parallèlement, les dynamiques communautaires ouvrent la voie à des alternatives open source renforcées (on l'a vu avec OpenTofu, ou l'omniprésence d'OpenTelemetry), assurant un écosystème ouvert et interopérable à long terme. L'**IA** s'infiltre progressivement dans le cycle DevOps, que ce soit via l'assistance au codage d'infrastructure, l'accélération des analyses de sécurité ou l'optimisation intelligente des ressources. Toutefois, cette puissance vient avec son lot de **nouveaux risques**, obligeant à élever le niveau de vigilance sur la chaîne logicielle (d'où l'importance grandissante du DevSecOps). Pour les équipes DevOps, les impacts à moyen terme seront multiples : adoption de ces nouvelles versions outillées (ex : planifier la montée en version Kubernetes 1.33 dans les prochains mois pour bénéficier des sidecars natifs), intégration des bonnes pratiques de sécurité (ex : mise en place de la signature d'images container si ce n'est pas déjà fait), et expérimentation des assistances IA pour gagner en productivité. L'horizon reste celui d'une **industrialisation toujours plus aboutie des workflows DevOps**, où la fiabilité, la rapidité et la sécurité vont de pair. Les annonces d'avril 2025 confirment que la communauté avance dans ce sens, en bâtissant sur des bases solides tout en innovant – de quoi alimenter les roadmaps internes des organisations cherchant à rester à la pointe de la transformation DevOps.

Sources :

- Changelog Jenkins 2.504.1 (LTS) – *Jenkins.io*
- Annonce GitLab 17.11 – *about.gitlab.com* (17 avril 2025)
- GitHub Changelog (15 avril 2025) – Nouveautés GitHub Actions
- Issue GitHub Actions Runner (mars 2025) – Support de Windows Server 2025
- CNCF Announcement (1er avril 2025) – Argo CD 3.0 RC, améliorations sécurité/perf
- Notes de version Terraform 1.12 (rc1) – *GitHub hashicorp/terraform*
- CNCF Project OpenTofu – Sandbox acceptance (23 avril 2025)
- Annonce Ansible 12 / Ansible-core 2.19 – *Ansible Forum*
- Pulumi Blog (avril 2025) – Nouveautés Pulumi (AI MCP, Comp. cross-langage, etc.)
- Article InfoQ – Kubernetes 1.33 *Octarine* (30 avril 2025)
- Article *Medium* LiveWyer – OpenTelemetry à KubeCon EU 2025
- Blog OpenTelemetry – *OpenTelemetry Logging and You* (18 avril 2025)
- The Hacker News – Alerte CVE-2025-30066 (19 mars 2025)
- JFrog BusinessWire – *Supply Chain Security in AI era* (1er avril 2025)
- Datadog *State of DevSecOps 2025* – Faits saillants (avril 2025)