

Mis à jour le 26/06/2025

S'inscrire

Formation LitmusChaos

3 jours (21 heures)

Présentation

Notre formation LitmusChaos vous permettra d'introduire de manière contrôlée des défaillances dans un système afin d'évaluer sa résilience et d'en améliorer la robustesse. Grâce à LitmusChaos, les équipes DevOps peuvent valider que leurs applications cloud-native résistent correctement aux pannes, aux perturbations réseau ou aux stress systèmes.

Notre formation LitmusChaos vous permettra d'apprendre à injecter, planifier et analyser des tests de chaos dans Kubernetes. Vous découvrirez comment utiliser les composants clés comme le ChaosEngine, le ChaosHub, et le portail ChaosCenter, et comment orchestrer des scénarios de perturbation simples ou avancés.

À l'issue de cette formation, vous serez en mesure de concevoir et exécuter des scénarios de chaos sur vos applications Kubernetes, d'en tirer des métriques exploitables, et d'automatiser vos tests de résilience.

Comme toutes nos formations, elle se déroulera sur la dernière version de l'outil : [Litmus 3.19](#)

Objectifs

- Comprendre les principes fondamentaux de la chaos engineering et leur intérêt pour la résilience applicative
- Installer et configurer LitmusChaos sur un cluster Kubernetes mono-cluster (local ou cloud)
- Déployer des expériences de chaos ciblées à l'aide des ressources Kubernetes (ChaosEngine, ChaosExperiment, ChaosResult)
- Concevoir, exécuter et analyser des scénarios de perturbation (CPU, mémoire, réseau, redémarrage de pods, etc.)
- Orchestrer plusieurs expériences via des Chaos Workflows
- Intégrer les tests de chaos dans un pipeline CI/CD et amorcer une démarche de Continuous Chaos

Public visé

- Ingénieurs DevOps
- SRE
- Développeurs back-end ou full-stack

Pré-requis

- Maîtriser les fondamentaux de Kubernetes (déploiement d'applications, gestion de pods/services, usage de kubectl).
- Avoir une expérience de base en DevOps ou administration de systèmes cloud-native.

Pré-requis techniques

- Disposer d'un poste de travail capable de faire tourner un cluster Kubernetes local ou d'un accès à un cluster distant

Programme de notre formation LitmusChaos

Introduction à la chaos engineering

- Définition et objectifs de la chaos engineering
- Résilience applicative et tolérance aux pannes
- Histoire et principes fondamentaux du chaos (steady-state, hypothèse, validation)
- Bonnes pratiques et cadre de sécurité pour les tests de chaos
- Présentation des outils du marché (Gremlin, Chaos Mesh, LitmusChaos)
- Positionnement de LitmusChaos dans l'écosystème CNCF

Architecture et composants de LitmusChaos

- Présentation générale de LitmusChaos (open source, CNCF, mono-cluster)
- Plan de contrôle vs plan d'exécution
- Les Custom Resources (ChaosEngine, ChaosExperiment, ChaosResult)
- Le Chaos Operator et le ChaosCenter
- Présentation du ChaosHub : la bibliothèque d'expériences
- Fonctionnement et cycle de vie d'une expérience de chaos

Mise en place de l'environnement LitmusChaos

- Préparation de l'environnement Kubernetes
- Installation de LitmusChaos (via Helm ou manifests YAML)
- Vérification des composants installés
- Déploiement d'une application de test (ex: Podtato-head, Nginx, etc.)
- Configuration des permissions et annotations nécessaires
- Accès au portail ChaosCenter (si utilisé)

- Atelier pratique : Installation de LitmusChaos et validation via un premier test simple (pod-delete)

Premier test de chaos avec ChaosEngine

- Structure et rôle d'un ChaosEngine
- Référencement d'un ChaosExperiment dans un ChaosEngine
- Définition de la cible via labels/selectors
- Paramétrage du test (durée, force, randomization, etc.)
- Exécution de l'expérience et lecture du ChaosResult
- Observation des effets sur l'application

Expériences de chaos classiques

- Types de chaos supportés par Litmus (CPU Hog, Memory Hog, Network Delay...)
- Pannes de pods, de containers ou de nœuds
- Dégradations réseau : latence, perte de paquets
- Stress de ressources : surcharge CPU, mémoire, disque
- Échecs de service dépendant (ex : base de données, DNS)
- Choisir le bon test selon la nature de l'application cible
- Atelier pratique : Création et exécution d'un ChaosEngine personnalisé (ex. : CPU Hog sur un microservice)

Observation et interprétation des résultats

- Analyse des logs d'exécution
- Utilisation de kubectl pour diagnostiquer les effets
- Consultation des objets ChaosResult
- Monitoring applicatif pendant l'expérience (Prometheus, Grafana, logs)
- Métriques clés de résilience à observer
- Débriefing une expérience pour en tirer des leçons

Orchestration avec les Chaos Workflows

- Introduction aux Chaos Workflows (enchaînement de tests)
- Construction visuelle via ChaosCenter ou définition YAML
- Dépendances et enchaînements conditionnels
- Exemple de workflow à deux étapes (pod-delete + memory-stress)
- Exécution et suivi d'un workflow
- Score de résilience global d'un workflow
- Atelier pratique : Conception et exécution d'un Chaos Workflow multi-étapes

Études de cas et planification du chaos

- Étude de cas : chaos engineering sur une application e-commerce
- Identification des points faibles à tester
- Choix progressif des expériences de chaos

- Organisation des campagnes de test (manuel, planifié, automatique)
- Stratégie de montée en charge des tests dans un contexte réel
- Établir un plan de chaos engineering d'équipe

Intégration de LitmusChaos dans la CI/CD

- Objectifs du chaos en pipeline CI/CD
- LitmusChaos et GitHub Actions, GitLab CI, Jenkins
- Déclenchement automatique des tests de chaos
- Validation et score de résilience en pipeline
- Exemples de configuration YAML pour CI/CD
- Vers le continuous chaos avec GitOps et ArgoCD

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.