Engineer

Alerte

un message système quand une personne doit intervenir

Ce que c’est

Une alerte est un signal automatique indiquant qu’une condition surveillée est restée vraie assez longtemps pour justifier d’interrompre quelqu’un. Certaines équipes parlent d’alarme ; une alerte assez forte pour réveiller la personne d’astreinte est généralement appelée page dans le vocabulaire anglophone.

Le mot couvre deux objets différents, que Prometheus sépare clairement. Une règle d’alerte Prometheus produit des alertes : une par ensemble distinct d’étiquettes pour lequel l’expression renvoie un résultat. Ces alertes vont vers un programme distinct, Alertmanager, qui les transforme en notifications : courriel, message de discussion, appel à un service d’astreinte. Le projet précise que les règles seules ne constituent pas un système complet de notification. La séparation compte, car les nombres divergent. Cinquante alertes peuvent arriver dans un message, une alerte produire un rappel toutes les quelques heures, et une alerte mise en sourdine n’atteindre personne tout en restant déclenchée.

Une alerte a un court cycle de vie. Tant que l’expression ne renvoie rien, elle est inactive. Au premier résultat, elle devient en attente et le reste pendant la durée demandée par la clause `for`. Si la condition reste vraie au terme de cette attente, elle se déclenche et est transmise pour livraison. Lorsque la condition cesse d’être vraie, elle est résolue. Une clause plus récente, `keep_firing_for`, maintient l’alerte déclenchée un moment après la disparition de la condition, pour prévenir, selon la documentation, les oscillations et les fausses résolutions. Prometheus consigne aussi les états en attente et déclenché dans une série temporelle nommée `ALERTS` ; les alertes elles-mêmes peuvent donc être interrogées et représentées.

Une règle qui attend vingt minutes avant de se déclencher, puis dix de plus avant de s’éteindre, ressemble à ceci :

```
groups:
    - name: exports
      rules:
          - alert: ExportQueueStalled
            expr: exports_waiting > 0 and rate(exports_done_total[15m]) == 0
            for: 20m
            keep_firing_for: 10m
            labels:
                severity: page
                team: billing
            annotations:
                summary: "{{ $value }} exports queued and none finishing"
```

Étiquettes et annotations se ressemblent, mais ont des fonctions distinctes. Les étiquettes identifient une alerte par rapport aux autres, et Alertmanager les lit pour choisir sa destination : `severity: page` peut atteindre un téléphone, `severity: ticket` une file de traitement. Les annotations apportent les mots nécessaires à la personne qui reçoit le message : résumé d'une ligne, description plus longue, lien vers la procédure indiquant la première action. Elles peuvent utiliser les étiquettes de l'alerte et la valeur déclenchante dans un modèle ; c'est ainsi que le résumé ci-dessus peut citer un nombre.

Alerte et incident sont deux mots distincts pour une raison. Une alerte signale qu'une condition était remplie ; déterminer si elle constitue un incident revient à une personne.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'une alerte ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Une supervision et des alertes qui signalent les incidents tant qu'ils restent mineurs, avec les chiffres qui montrent à quelle fréquence ils le restent.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Premier tableau de bord d'observabilité — première vue de l'entreprise sur ses propres systèmes

Premier tableau de bord d'observabilité Contactez-nous →

Pourquoi c'est important

Une alerte est la seule partie de la supervision qui n'attend pas d'être consultée. Une métrique reste stockée, un tableau de bord attend sur un écran, un journal attend une recherche, et une alerte s'invite dans la soirée de quelqu'un. Chaque règle devient donc une petite sollicitation permanente de son attention, justifiée

seulement si la personne peut agir. Les recommandations Prometheus conseillent ainsi d'éviter les appels d'astreinte sans action possible et de garder l'alerte simple.

La deuxième raison est arithmétique. Une cause unique déclenche généralement plusieurs règles. Lorsqu'une partition réseau coupe la moitié des instances d'un service de leur base, chacune peut déclencher sa propre alerte. La documentation Alertmanager utilise précisément ce cas pour transformer des centaines d'alertes en une notification. Alertmanager rassemble celles partageant les étiquettes de `group_by`, attend `group_wait` (30 secondes sans autre réglage) pour que le premier message contienne tout le lot, n'envoie les mises à jour du groupe qu'à intervalles d'au moins `group_interval` (5 minutes) et répète un message toujours actif après `repeat_interval` (4 heures). L'inhibition va plus loin : elle supprime certaines alertes tant qu'une autre est active, afin qu'une alerte sur une grappe entière inaccessible fasse taire celles de ses services. Les silences mettent en sourdine les alertes correspondant à un ensemble d'étiquettes pendant une période définie, généralement lors d'une maintenance planifiée.

La troisième raison est la fin de l'histoire. La plupart des destinataires envoient un message de résolution à la disparition de l'alerte, mais `send_resolved` vaut faux sans réglage explicite pour le destinataire courriel. Une équipe qui alerte par courriel peut donc voir le problème annoncé sans jamais en voir la fin.

À quoi cela ressemble en pratique

L'hôte de l'entreprise utilise une alarme par absence de signal. Ses 464 Mo de mémoire servent un serveur web et une forge, et l'épuisement de la mémoire ainsi qu'un disque plein sont deux risques de panne importants. Un temporisateur sur l'hôte contacte un service extérieur uniquement si ses contrôles des seuils de mémoire disponible et d'espace disque libre réussissent, toutes les quinze minutes avec une légère variation ; le service extérieur déclenche l'alarme lorsque les signaux attendus cessent d'arriver. Les pannes empêchant le script de s'exécuter ou de joindre ce service, dont une perte réseau ou un démarrage raté, produisent une absence de signal. La réception d'un signal confirme que le script s'est exécuté et que ses contrôles des seuils de mémoire disponible et d'espace disque libre ont réussi ; elle ne certifie ni le bon état général de l'hôte ni que le site sert des pages. La supervision peut révéler des signes avant-coureurs, dont le moment d'apparition dépend toutefois de la charge de travail et des conditions observées.

La même inversion peut protéger les tâches planifiées. La mise en œuvre des sauvegardes sépare les signaux de réussite de sauvegarde et de restauration. Ses scripts et unités systemd prennent en charge des instantanés chiffrés hors hôte, des contrôles d'intégrité et un test mensuel de restauration d'un fichier connu ; un signal de réussite n'est envoyé qu'après le succès du processus. C'est la conception du parcours d'alerte, pas la preuve de son activation. Le dossier de mise en œuvre daté du 2026-10-27 indique que les temporisateurs étaient désactivés sans dépôt configuré et ne consigne aucune restauration réelle achevée. Le silence peut déclencher une alerte extérieure une fois le processus et sa supervision activés ; la configuration seule ne démontre pas leur fonctionnement.

Les alertes inversées ont leur propre mode de défaillance, apparu au passage prévu d'un hôte à deux. Partager un seul dispositif entre les deux aurait affaibli les alertes qu'il était censé étendre : tant que l'un émet ses signaux, le contrôle reste vert alors que l'autre est mort. Chaque dispositif est devenu une valeur par hôte, et les entrées de l'hôte existant ont été délibérément laissées vides : la configuration indique ainsi que le raccordement existe, mais que la valeur reste à fournir.

Où les problèmes surviennent

Une valeur mesurée finit dans une étiquette. Les étiquettes identifiant l'alerte, y placer le nombre courant transforme chaque variation en nouvelle alerte et résout la précédente. Le destinataire reçoit une suite de paires déclenchement–résolution pour un problème unique. Réservez les étiquettes à l'identité et au routage, et les annotations aux nombres.

L'alerte oscille. Une condition proche du seuil se déclenche, se résout puis recommence toutes les quelques minutes, chaque résolution annonçant une information fausse. Allongez `for` pour qu'un incident passager disparaisse avant d'être signalé, et utilisez `keep_firing_for` pour qu'une baisse brève sous le seuil ne termine pas l'alerte.

Un silence survit à sa justification. Une personne coupe une alerte pour maintenance avec un filtre large et une fin lointaine ; la vraie panne de la semaine suivante arrive donc en sourdine. Limitez les silences au plus près du travail, fixez leur fin près de la fin prévue et consignez leur auteur et leur raison dans le champ de commentaire d'Alertmanager.

Une route large absorbe les routes précises. Alertmanager parcourt l'arbre de routage dans l'ordre et, sauf `continue: true`, s'arrête au premier enfant correspondant. Une route générale au-dessus de celle d'une équipe récupère ses alertes. Placez les routes les plus précises en premier et testez l'arbre avec des étiquettes d'exemple avant de lui faire confiance.

Tout est urgent. Disque à 70 %, certificat expirant le mois prochain et service de paiement en erreur appellent tous le même téléphone, bientôt ignoré. N'appelez la personne d'astreinte que pour une condition nécessitant une action immédiate, envoyez le reste dans une file de tickets ou une synthèse quotidienne, et supprimez toute règle déclenchée depuis des mois sans intervention.

Pour aller plus loin

- Vue d'ensemble des alertes — partage du travail entre règles Prometheus et Alertmanager, et étapes de leur configuration.
- Règles d'alerte — syntaxe, notamment `for`, `keep_firing_for`, étiquettes, annotations et modèles.
- Alertmanager — regroupement, inhibition, silences et exécution de plusieurs Alertmanager pour la haute disponibilité.
- Configuration d'Alertmanager — arbre de routage, temporisateurs de regroupement, règles d'inhibition et paramètres des destinataires, dont `send_resolved`.
- Bonnes pratiques d'alerte — conseils du projet sur ce qui mérite un appel d'astreinte et ce qui doit aller ailleurs.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'une alerte ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/alert/