ADVENT OF SRE

AM

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

ADVENT OF SRE

1

AM

SRE, c'est quoi ?

Le **Site Reliability Engineer** (SRE) est l'ingénieur de la fiabilité des sites (en bon français) 🇫🇷

Son objectif principal ? S'assurer que les applications et infrastructures soient **fiables, performantes et disponibles** pour les utilisateurs 💪

Ses missions ?*

Il s’assure que les services respectent les **engagements de disponibilité** (SLOs & Error Budget) 🤝

Il fait en sorte que les applications **fonctionnent** de manière **fluide** et ce, même en cas de pic de trafic 📈

* Ici, j’ai evoqué les principales missions. Mais il en a d’autres, on aura l’occasion d’en reparler les prochains jours

ADVENT OF SRE 2

AM

SLI

Le Service Level Indicator (SLI) va **mesurer** la performance ou de la fiabilité d'un service 📊

Ça peut être un taux de dispo, une latence moyenne ou un taux d'erreur.

Du style *"99.9% des requêtes doivent se faire en moins de 300ms"*

SLO

Le Service Level Objective est le **niveau de service** que l'on s'engage à **respecter** 🤝

Si notre SLI est "*un taux de succès des requêtes*", notre SLO pourrait être "*99.5% de succès*"

SLA

Le Service Level Agreement est un **contrat** qui définit le **niveau de dispo** ou de performance **garanti** 📋

Mais aussi les **conséquences** en cas de **non-respect**. Un SLA pourrait indiquer une disponibilité de 99.9% par mois

Pourquoi ?

Les **SLOs** et **SLIs** permettent de définir ce qui est acceptable et ce qui ne l'est pas 

Ca aide à trouver un **équilibre** entre **développement et stabilité** (tiens, ça rappelle pas un peu DevOps ? 🤔)

ADVENT OF SRE

3

AM

Un bon SLI ?

Doit être **pertinent**. Comme la latence ou le taux d'erreur 💪

L'indicateur doit permettre de **prendre des décisions**. Si le taux de succès baisse, on sait qu'il y a un problème 📉

AM

Mesurer les SLIS ?

On peut utiliser des outils pour **collecter** les métriques 🧮

Configurer des outils de **logging** et de **tracing** pour analyser les requêtes

Créer des dashboards pour **visualiser** nos SLIs et configurez des alertes en cas de besoin 📊

AM

ADUENT OF SRE

4

AM

Un error budget ?

C’est la **quantité** de "problèmes" que l’on peut se **permettre** avant de marcher sur son SLO 👈

Du style, notre SLO c’est une dispo de 99.9% sur un mois donc on a le “droit” à 0.1% de temps d’indisponibilité

Pourquoi ?

On peut **tester** de nouvelles features, faire des déploiements plus **rapides**, plus **souvent,** accepter un peu de risque tant que le budget n'est pas épuisé 🧪

S'il est consommé rapidement (trop d'incidents), on va **ralentir** et faire des **corrections** ou de l'optimisation

Comment l'utiliser ?

Il faut fixer un SLO clair. Genre, une disponibilité de 99.9% pour un service critique 

Ensuite, on va avoir des outils de monitoring pour mesurer le taux d'utilisation 

AM

ADVENT OF SRE

5

AM

Le monitoring ?

Ça consiste à **collecter**, **visualiser** et **alerter** sur des métriques connues

🔑 Le monitoring répond à la question "*Le système fonctionne-t-il comme prévu ?*"

AM

L'observabilité ?

Elle consiste à **instrumenter** les systèmes pour **comprendre** ce qui ne va pas

🔑 L'observabilité répond la question "*Pourquoi ce système ne fonctionne-t-il pas comme prévu ?*"

Les différences ?

Le **monitoring** vous dit que "*la latence a augmenté*"

L'observabilité vous aide à comprendre pourquoi la latence a augmenté

AM

ADVENT OF SRE

6

AM

Metrics ?

Les métriques sont des **mesures** dans le temps, souvent utilisées pour suivre l'état de santé d'un système/appli 📊

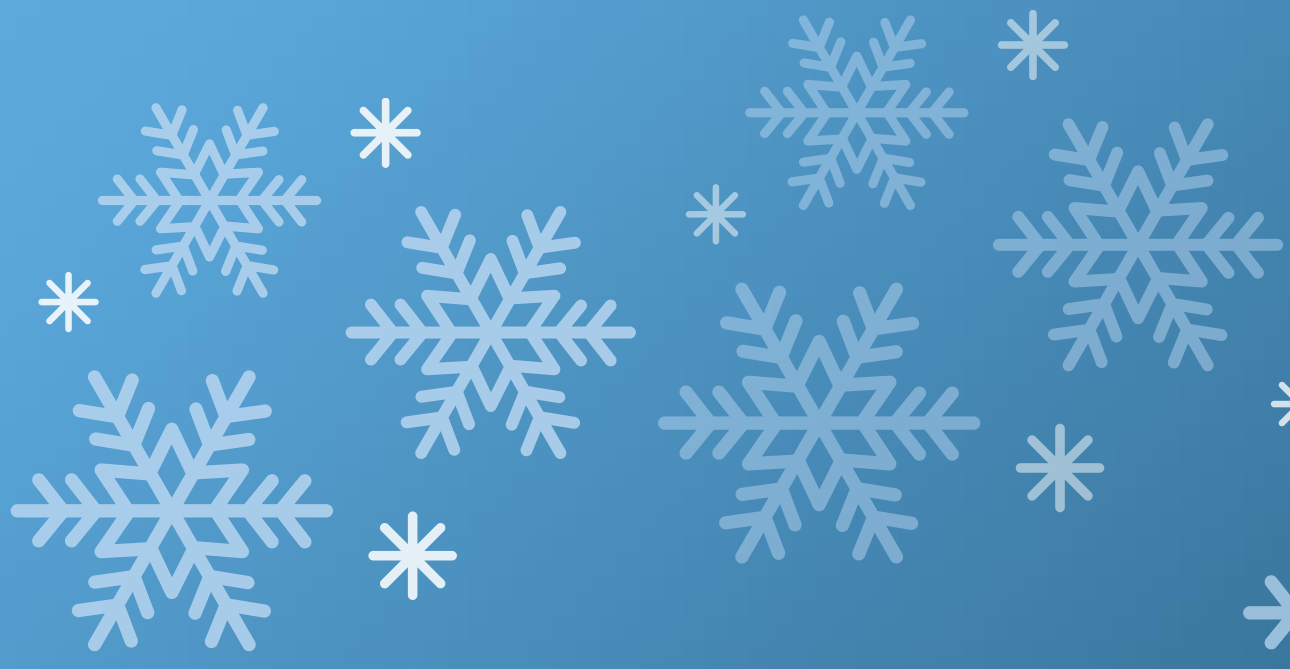
Elles fournissent une **vue d'ensemble** et elles déclenchent souvent des alertes en cas de dépassement de seuils critiques 🚨

AM

Logs ?

Ce sont des enregistrements détaillés des **événements** se produisant dans les systèmes/appli 📝

Ils permettent **d'analyser** un problème 🧐



Traces ?

Les traces suivent le **parcours d'une requête** à travers plusieurs services ou composants 🕵️

Elles permettent de visualiser les **dépendances** entre services et elles aident à identifier où un problème se produit (goulot d'étranglement, service lent...)

ADVENT OF SRE

7

AM

Prometheus ?

C’est un outil qui **collecte** et **stock** des **métriques** en temps réel 🎣

Il les récup’ depuis des **endpoints** exposés par les **services** 👋

Il peut suivre la latence, l’utilisation CPU ou le taux de succès des requêtes

Grafana ?

C’est un outil de **visualisation** qui permet de **créer** des **dashboards** 📊

Avec cet outil on peut **suivre** nos métriques, logs, traces avec des graphiques, jauges et tableaux 💻

AM

Pourquoi ?

1️⃣ Tous les deux ont une large communauté

2️⃣ Ils s'intègrent parfaitement, Prometheus collecte les métriques, Grafana les visualise

AM

ADVENT OF SRE

8

AM

Les bases ?

Une alerte doit être **actionnable**. Elle n'a de sens que si elle nécessite une **intervention** (automatique ou humaine) 🛠️

On n'a pas d'alertes si l'on a trop d'alertes !

Bonnes pratiques ?

1️⃣ **Priorisez** les alertes qui ont un impact

2️⃣ Évitez les seuils **trop sensibles**

3️⃣ Ajoutez un **contexte clair** (quel service, nature du problème…)

4️⃣ **Testez** régulièrement vos alertes

Anti-Patterns ?

🚫 Le "**Tout alerter**", surveiller chaque détail produit de la fatigue et réduit la réactivité

🚫 Les alertes **non actionnables**, "*Si une alerte ne nécessite aucune action immédiate, elle n'est pas une alerte.*" Faites des dashboards, mais ne spammez

ADVENT OF SRE

9

AM

Pourquoi ?

Lorsqu'il y a un incident, le **stress** et la **pression** augmentent 😱

Sans une **bonne préparation**, on arrive à des erreurs humaines ou une mauvaise communication ce qui va **augmenter** le temps de résolution et **l'impact** sur les utilisateurs 👥

AM

Des acteurs ?

Une équipe avec un **Incident Commander** qui va coordonner la réponse et décider des priorités 👈

Un **scribe** pour documenter les actions en temps réel et des **Experts techniques** qui vont diagnostiquer et résoudre les problèmes 👨‍💻

Les bonnes pratiques ?

1️⃣ Simulez des incidents régulièrement

2️⃣ Centralisez les informations

3️⃣ Mettez à jour vos documents

4️⃣ Encouragez une culture blameless

ADVENT OF SRE

10

AM

Pourquoi ?

Un post-mortem **documente** l'incident pour permettre à toute l'équipe de **comprendre** ce qui s'est passé 📖

À noter qu'un bon post-mortem se concentre sur les **faits** et les **solutions**, pas sur les individus

La structure ?

1. Résumé de l'incident
2. Impact
3. Chronologie
4. Root Causes
5. Actions correctives
6. Enseignements clé

Bonnes pratiques ?

Un post-mortem ne doit pas chercher à pointer du doigt des **erreurs humaines** 👊

Partagez-les avec toutes les parties prenantes (même au-delà) 🔗

Chaque post-mortem doit déboucher sur des **actions concrètes** 💪

AM

ADVENT OF SRE

11

AM

Pourquoi ?

Les tâches **répétitives** prennent du temps, automatiser permet de se concentrer sur l'**innovation** plutôt que sur la maintenance 🛠️

On limite les **incidents** liés à des erreurs de configuration **manuelles** ou à des oublis 💭

AM

Par où commencer ?

Par les tâches qui consomment beaucoup de **temps** ou causent fréquemment des **erreurs** ⏳

Commencez par automatiser d'abord des **tâches simples** et peu risquées

Choisissez des outils **standardisés** et reconnus (Terraform, Ansible, Python/Bash...) 💪

ADVENT OF SRE

12

AM

Infra As Code ?

L'IaC consiste à **gérer** et **provisionner** l'infrastructure avec du code 👨‍💻

On va **décrire** l'état **désiré**, et l'outil se charge de l'atteindre 🎯

Le best, c'est de faire comme votre code applicatif, le **stocker** dans un **dépôt git**

AM

Cas d'usage ?

1️⃣ Création d'un environnement Kubernetes

2️⃣ Déploiement multicloud

3️⃣ Sandbox pour développeurs

Des Outils ?

👉 Terraform

👉 AWS CloudFormation

👉 Pulumi

👉 Ansible

ADVENT OF SRE

13

AM

Terraform ?

Développé par HashiCorp, c'est un outil **déclaratif** pour le provisionnement d'infrastructures 🛠️

En gros, on défini l'**état final** (infrastructure désirée), et Terraform se charge du reste 😎

AM

Ansible ?

Développé par RedHat, c'est un outil d'automatisation pour la gestion de la **configuration**, le **déploiement** et l'**orchestration** 🛠️

Très utile pour l'**automatisation** de **tâches** sur des serveurs, des **conteneurs** ou des **applications** 🌍

AM

Ansible ou Terraform ?

Et pourquoi pas les deux ?

1️⃣ Terraform **crée** les instances (machines virtuelles, bases de données, réseaux...)

2️⃣ Ansible **configure** les services ou déploie des applications sur ces instances

ADVENT OF SRE

14

AM

Un conteneur ?

C'est un **"package"** léger, portable qui regroupe généralement une appli avec ses **dépendances**, et tout ce dont elle a besoin pour fonctionner dans un env' isolé 📦

AM

Le concept ?

Une **image** est **immuable** et contient tout ce qu'un conteneur à **besoin** (un serveur web Nginx configuré) 🤳

Un **conteneur** est une **instance** en **cours d'exécution** d'une image Docker 🚚

ADVENT OF SRE

15

AM

Pourquoi ?

1️⃣ **Automatiser** le déploiement

2️⃣ Assurer la **résilience**

3️⃣ **Scalabilité** automatique

4️⃣ **Simplifier** les mises à jour

Concepts clés ?

1️⃣ **Pods** : Le plus petit objet déployable

2️⃣ **Services** : Expose les pods pour permettre leur accès

3️⃣ **Deployments** : Décrivent comment déployer et gérer vos pods

ADVENT OF SRE

16

AM

Cas pratique ?

Prenons une application web

1️⃣ Un **Pod** pour héberger un serveur Nginx
2️⃣ Un **Service** pour exposer le serveur
3️⃣ Un **Deployment** pour s'assurer qu'au moins 3 Pods sont actifs à tout moment

ADVENT OF SRE

17

AM

Une probe ?

Mécanisme utilisé par Kube pour interroger **périodiquement** l'état de **santé** d'un conteneur. Elle permet de répondre à 2 questions :

👉 Mon application fonctionne bien ?

👉 Est-elle est prête à recevoir du trafic ?

Les types ?

1️⃣ **Liveness Probe :** Est-ce que l'application est vivante ?

2️⃣ **Readiness Probe :** Est-ce que l'application est prête ?

Pourquoi ?

👍 Augmenter la **résilience**

👍 Garantir la **fiabilité**

ADVENT OF SRE

18

AM

La CI ?

CI (Intégration Continue)

👉 **Automatisation des tests :** Valider le code après chaque modification/commit

👉 **Tests d'intégration :** Garantir que les changements s'intègrent correctement dans le code existant

La CD ?

CD (Déploiement Continu)

👉 **Déploiement Continu :** Les changements validés sont automatiquement déployés

Des outils CI/CD ?

1. GitLab CI/CD
2. GitHub Actions
3. ArgoCD et FluxCD
4. Jenkins

Les avantages ?

Une CI/CD bien conçue permet de :

👍 **Réduire** le Time-to-Market

👍 **Diminuer** les erreurs humaines

👍 **Augmenter** la fiabilité

ADVENT OF SRE

19

AM

Pourquoi ?

Dans un cluster Kube :

👉 Suivre la **performance** des Pods et des nœuds

👉 Diagnostiquer les **problèmes** rapidement

Metric Server ?

Le Metrics Server est une **extension** native Kubernetes qui **collecte** les **métriques** des ressources comme, l'utilisation du **CPU** ou de la **mémoire** des Pods et des nœuds ⛏️

Promtheus Operator ?

Prometheus est un **outils** pour le **monitoring**. Avec le Prometheus Operator, l'**intégration** dans Kubernetes est **plus facile** 💪

Les bénéfices ?

👍 **Diagnostiquer** rapidement les problèmes dans un cluster

👍 **Optimiser** l'utilisation des ressources en détectant les surcharges ou sous-utilisations

ADVENT OF SRE

20

AM

Les fondamentaux ?

Minimiser les vecteurs d'attaque en limitant la **surface d'exposition**

Donnez à chaque utilisateur ou service uniquement les **permissions nécessaires** 🛡️

Chiffrer vos données (au repos, en transit) 🔒

AM

Bonnes pratiques ?

1. **Mettez à jour** régulièrement
2. **Surveillez** les journaux
3. Effectuez des **audits** de sécurité
4. **Sensibilisez** vos équipes
5. **Planifiez** la réponse aux incidents

ADVENT OF SRE

21

AM

Pourquoi ?

1️⃣ Permet d'appliquer le **principe du moindre privilège**

2️⃣ **Limiter les actions** que les users, services, ou envs' peuvent effectuer

AM

Concepts ?

1️⃣ **Rôle** : un ensemble de permissions appliqué à un namespace

2️⃣ **ClusterRole** : Comme un Role, mais applicable au cluster entier

3️⃣ **Binding** : Associe un utilisateur/groupe à un Role ou ClusterRole

ADVENT OF SRE

22

AM

Scalabilité ?

C'est la capacité d'un système à **s'adapter** à une augmentation/diminution de la charge

👉 **Verticale** : Ajouter ou retirer des ressources

👉 **Horizontale** : Ajouter ou retirer des instances

AM

Bonnes pratiques ?

Si vous avez des pics prévisibles,
préparez vos systèmes en amont
avec des capacités supplémentaires

Utilisez un **auto-scaler** ⚙️

Répartissez efficacement le trafic
avec des **load balancers** 🧮

AM

ADVENT OF SRE

23

AM

Pourquoi ?

Identifier les ressources inutilisées ou surprovisionnées 🧐

Réserver les **budgets** pour des initiatives à **forte valeur ajoutée** 💪

Maintenir des **services fiables** tout en respectant les contraintes financières 

AM

Bonnes pratiques ?

1️⃣ Adaptez les tailles des instances

2️⃣ Utilisation de Spot Instances

3️⃣ Auto-Scaling & Shutdown Auto

4️⃣ Optimisation du Stockage

5️⃣ Suivi et Allocation des Coûts

ADVENT OF SRE

24

AM

Conseils ?

1️⃣ Maîtrisez les **fondamentaux**

2️⃣ Créez des **projets perso**

3️⃣ **Apprenez** à gérer les incidents

4️⃣ Développez votre esprit de **Collaboration**

Mon Astuce ultime ?

Ne cessez jamais d'apprendre !

Le domaine **SRE** est en constante **évolution** :

👉 Suivez des blogs

👉 Écoutez des podcasts

👉 Participez à des conférences