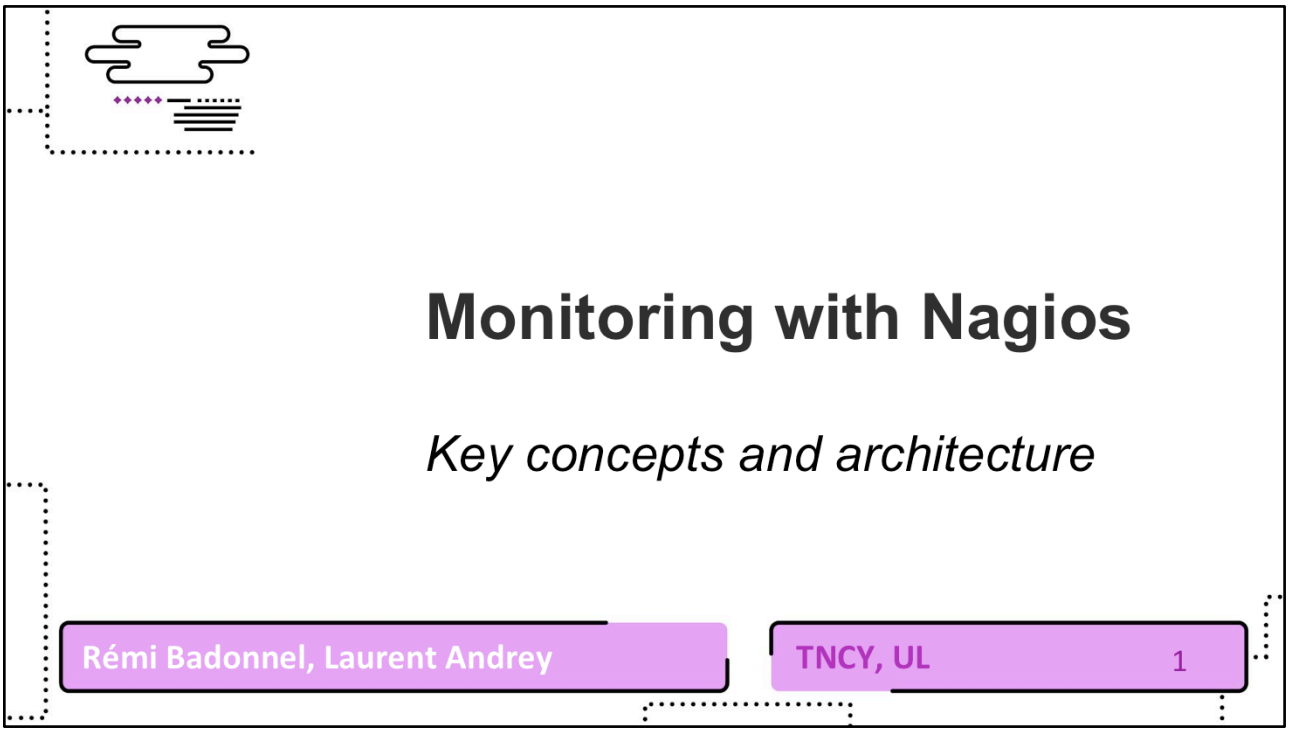
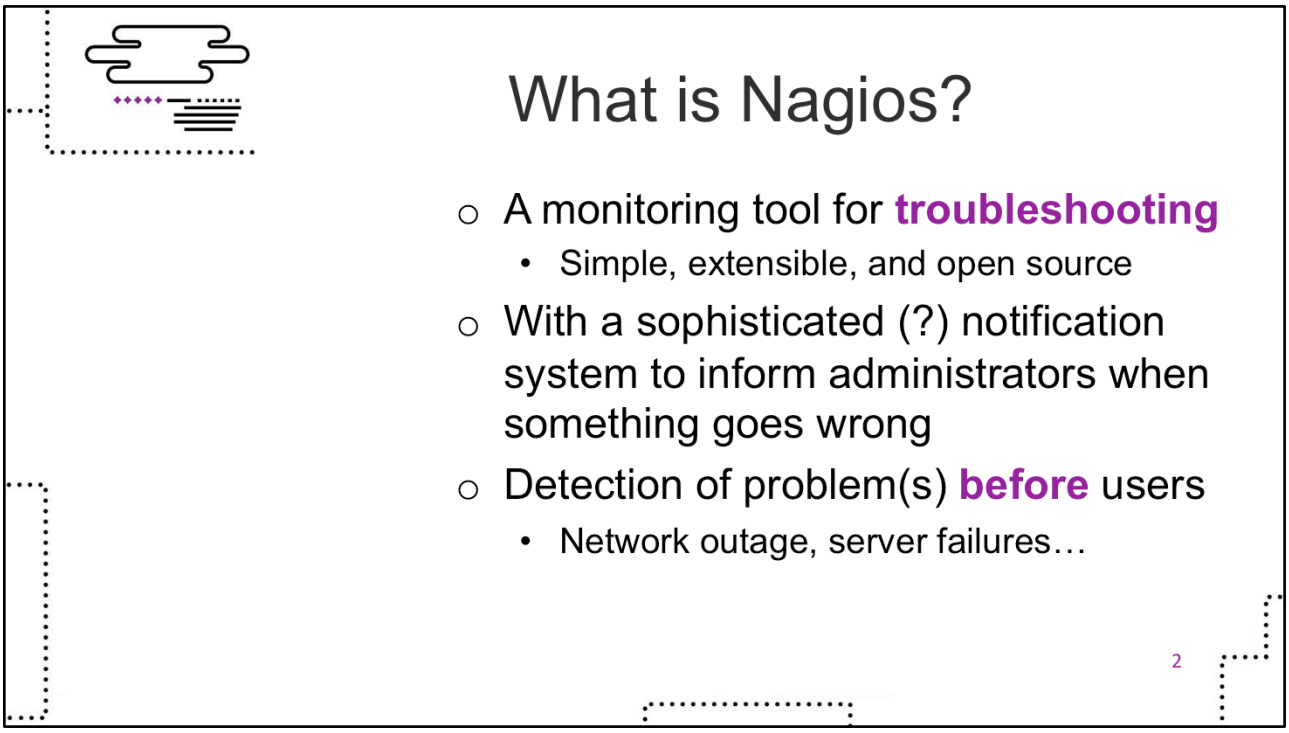


Bonjour à tous ! Cette semaine, nous allons aborder la pratique du monitoring avec l'outil Nagios. Cette première leçon va nous permettre de découvrir les concepts clés de l'outil et de détailler son architecture fonctionnelle.



Tout d'abord, à quoi correspond Nagios et quels en sont les usages ?

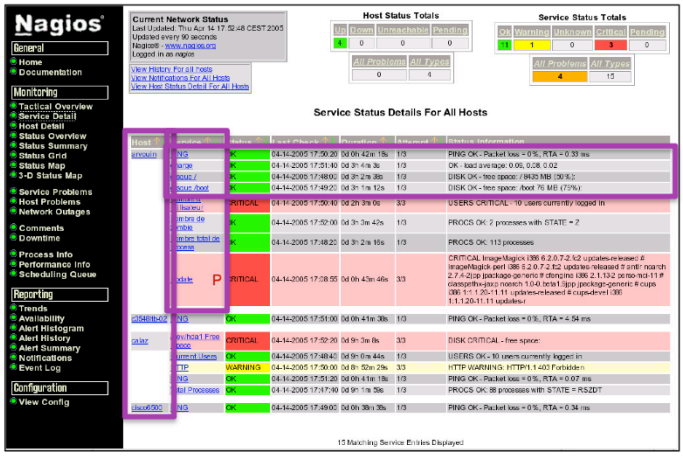
Nagios est un outil de monitoring largement répandu pour assurer la détection de pannes ou plus généralement de dysfonctionnements au sein d'une infrastructure réseau. Si l'on se réfère aux aires fonctionnelles de la gestion de réseaux, on le placera donc plutôt dans la catégorie de la gestion des fautes plutôt que celle de la gestion de performances, même si sur certains aspects il peut y contribuer également.

On peut qualifier l'outil de simple, extensible et open source. Simple car vous allez pouvoir facilement déployer l'outil sur votre propre infrastructure. Extensible car vous pourrez créer et ajouter des plugins spécifiques pour répondre à vos propres besoins. Et enfin, open source car c'est un logiciel libre qui est fourni sous licence GPL.

Nagios intègre un système de notifications relativement sophistiqué, qui permet d'avertir les administrateurs lorsqu'une panne ou un dysfonctionnement est détecté. On entend par sophistiqué, un système de notifications, qui intègre des mécanismes de filtrage et d'escalade de notifications. Le filtrage va permettre de s'assurer que l'on envoie la bonne information à la bonne personne et surtout d'éviter la génération d'un trop grand nombre d'alertes. Le mécanisme d'escalade de notifications va, quant à lui, permettre d'établir une hiérarchie des contacts parmi une liste d'administrateurs. Typiquement, lorsqu'une panne va être détectée et qu'une première notification a été envoyée, si la panne n'a pas été résolue par le premier contact, alors Nagios va commencer à informer un contact de niveau supérieur dans la hiérarchie.

L'objectif de Nagios est clairement de venir en soutien aux administrateurs pour identifier les problèmes avant que les usagers de l'infrastructure ne s'en rendent compte eux-mêmes et ne soient touchés par ceux-ci. Pour ce faire, il s'appuie sur des batteries de tests qui sont exécutés régulièrement sur l'infrastructure. Nagios peut être utilisé pour détecter des problèmes de nature diverse, tels que par exemple une panne réseau, la défaillance d'un serveur de mail, ou la surcharge d'un serveur de stockage. Il peut aussi être utilisé pour contrôler des services qui peuvent être externes à votre infrastructure, tels que des services qui sont fournis par le cloud.

How does it look like?

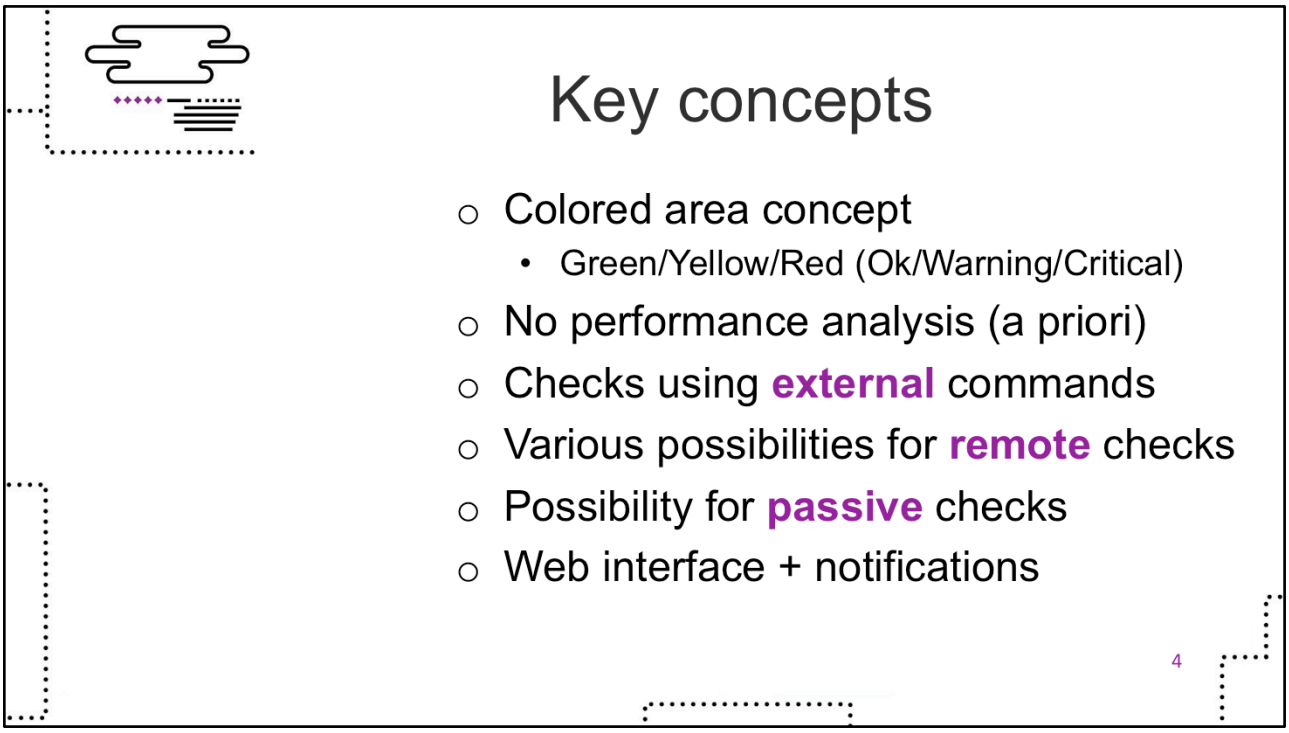


3

Sous quelle forme se présente Nagios ?

Cette slide fournit un aperçu de son interface web. Celle-ci décrit une liste de machines (ou hosts) incluant serveurs et routeurs qui sont surveillés par Nagios. Pour chacune de ces machines, Nagios fournit une liste de services qui sont testés par l'outil.

La notion de services est entendue au sens large sous Nagios. Cela correspond à tout paramètre ou service qui peut être observé sur un équipement. Nagios fournit un état (ou status) pour chaque service. Cet état indique si le service fonctionne correctement ou non à l'aide d'un code couleur. Il est complété par des informations relatives aux tests, telles que la dernière fois qu'un test a été exécuté, le nombre de tests qui ont été exécutés jusqu'à présent, ou encore des informations détaillées sur les résultats des tests.



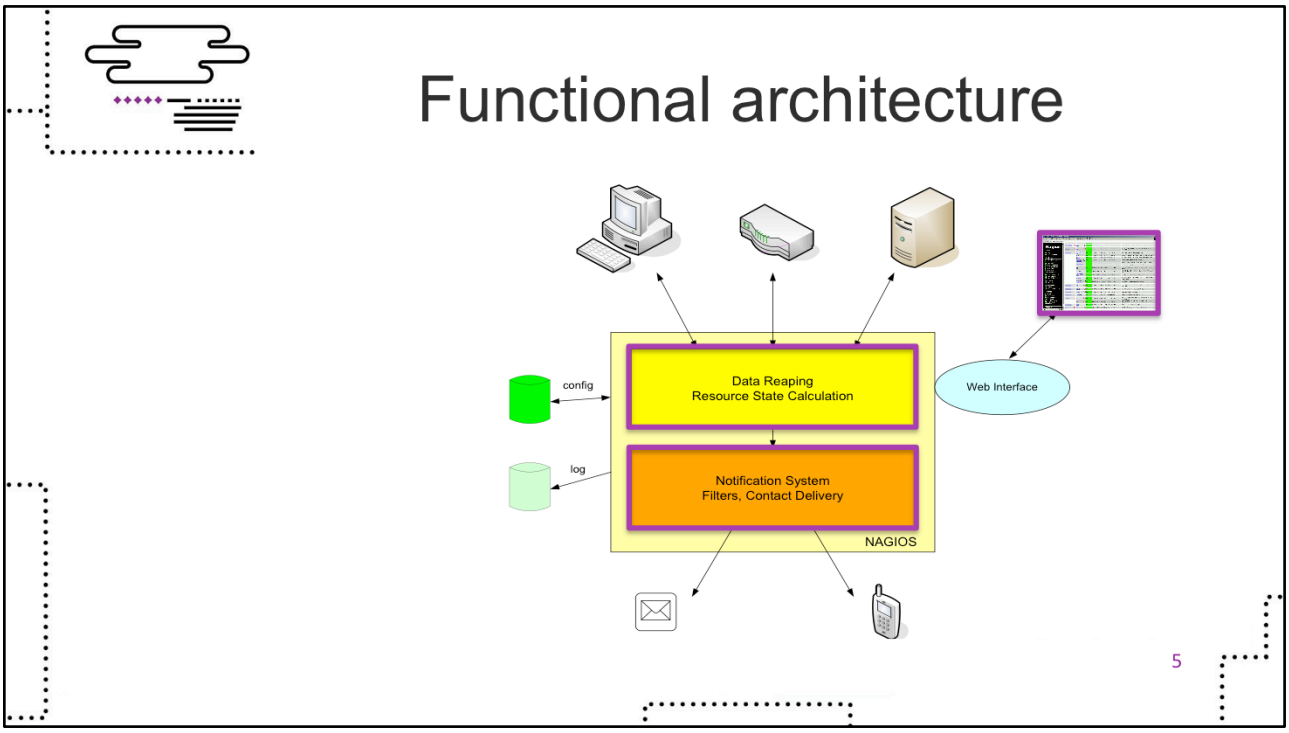
Nagios repose sur plusieurs concepts clés (décrits ci-dessous).

Un premier concept important est celui de code couleur. L'état d'un service est caractérisé à chaque fois par une couleur : le vert indique que le service est opérationnel (ou OK), le jaune indique que celui-ci est dans un état d'alerte (ou WARNING), et le rouge qu'il a atteint un état critique (également appelé CRITICAL). Le sens, qui est donné aux états WARNING, CRITICAL et OK, est défini par l'administrateur et peut varier en fonction du scénario d'usage. Comme mentionné précédemment, Nagios n'a pas été conçu au départ comme un outil pour faire de l'analyse de performances. On peut néanmoins noter qu'il existe des plugins dédiés à cela, en particulier les plugins fournis par Cacti.

Un second concept important est celui de tests (ou checks). Les tests sont exécutés périodiquement par le serveur Nagios, afin d'inférer l'état des services. Un test est implanté à l'aide d'un plugin, et un plugin correspond typiquement à une commande ou un script, qui est exécuté sur un équipement.

Nagios fournit différentes façons d'exécuter les tests de manière distante, telles qu'en utilisant ssh ou un agent dédié, qui est l'agent Nagios NRPE. Il est également possible d'exécuter des tests de manière passive, en s'appuyant sur un concept similaire à celui de traps SNMP. Dans ce cas, le serveur Nagios ne va pas périodiquement faire des requêtes auprès des agents. Mais les agents vont uniquement rapporter des alertes au serveur, lorsque certains événements ont lieu : par exemple, en cas de dépassement d'une valeur seuil.

Les administrateurs interagissent avec Nagios via son interface Web, mais aussi grâce aux notifications qui leur sont envoyées par l'outil, lorsque des services changent d'états.



Nous décrivons ici l'architecture fonctionnelle de Nagios. Cette architecture se décompose en deux blocs principaux.

Un premier bloc appelé "Data Reaping" chargé de collecter les résultats des tests, mais aussi de calculer, à partir de ces résultats, l'état effectif des ressources : que ce soit l'état des machines ou l'état de leurs services.

Le second bloc correspond au système de notifications, qui est chargé d'envoyer les alertes aux administrateurs. Il intègre des mécanismes de filtrage et d'escalade de notifications. Les alertes sont envoyées aux administrateurs sous forme de mails ou de SMS.

Les deux blocs sont implantés sous la forme de processus Nagios, qui sont exécutés comme des services sur le système d'exploitation. Nagios fournit également une interface Web qui est implantée à l'aide d'un serveur web, et d'un ensemble de scripts cgi. La configuration quant à elle s'appuie typiquement sur de simples fichiers texte.