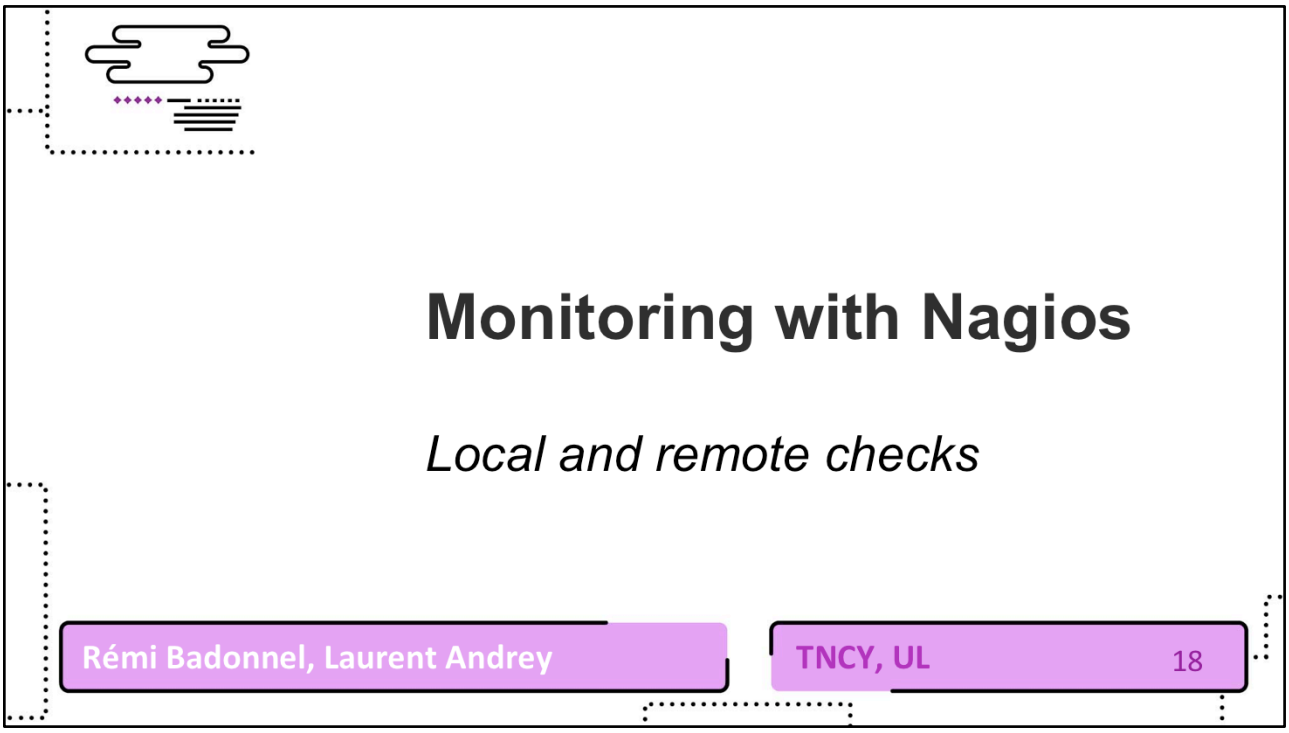
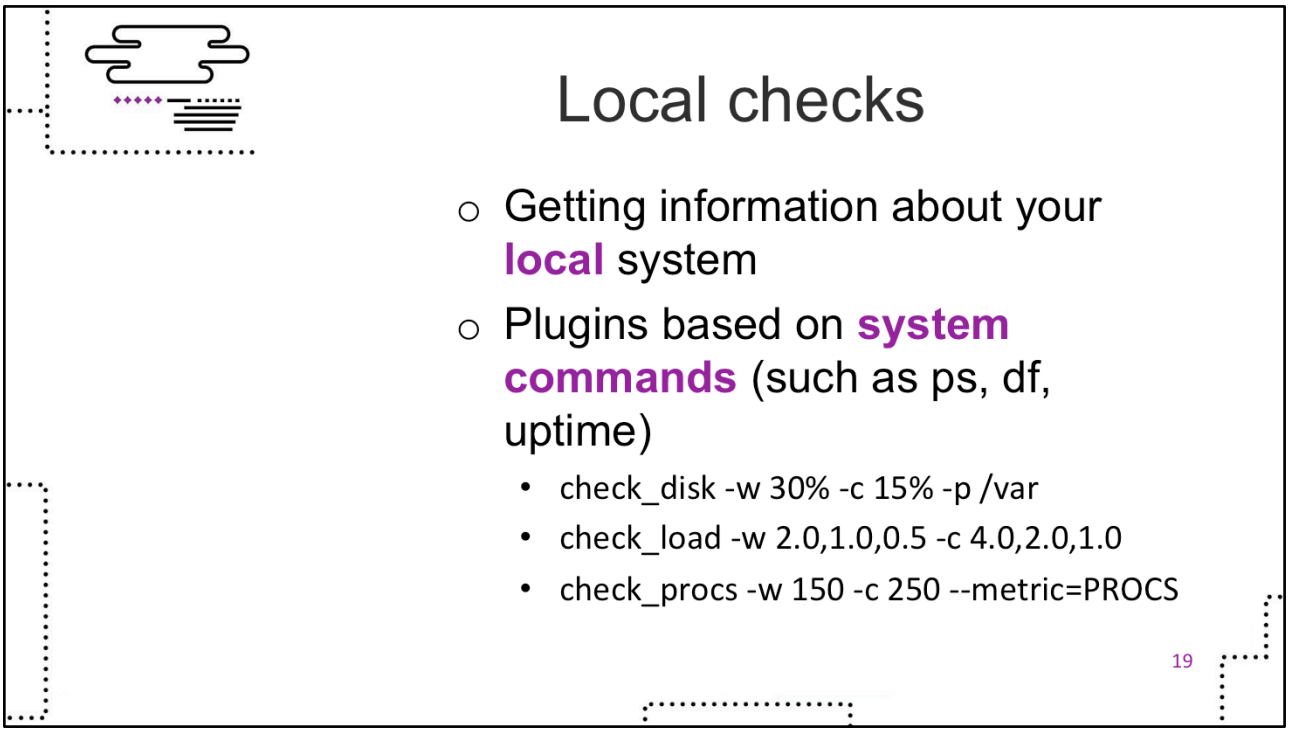


La leçon suivante va aborder les différents types de tests sous Nagios, et plus particulièrement les différentes façons de les exécuter.

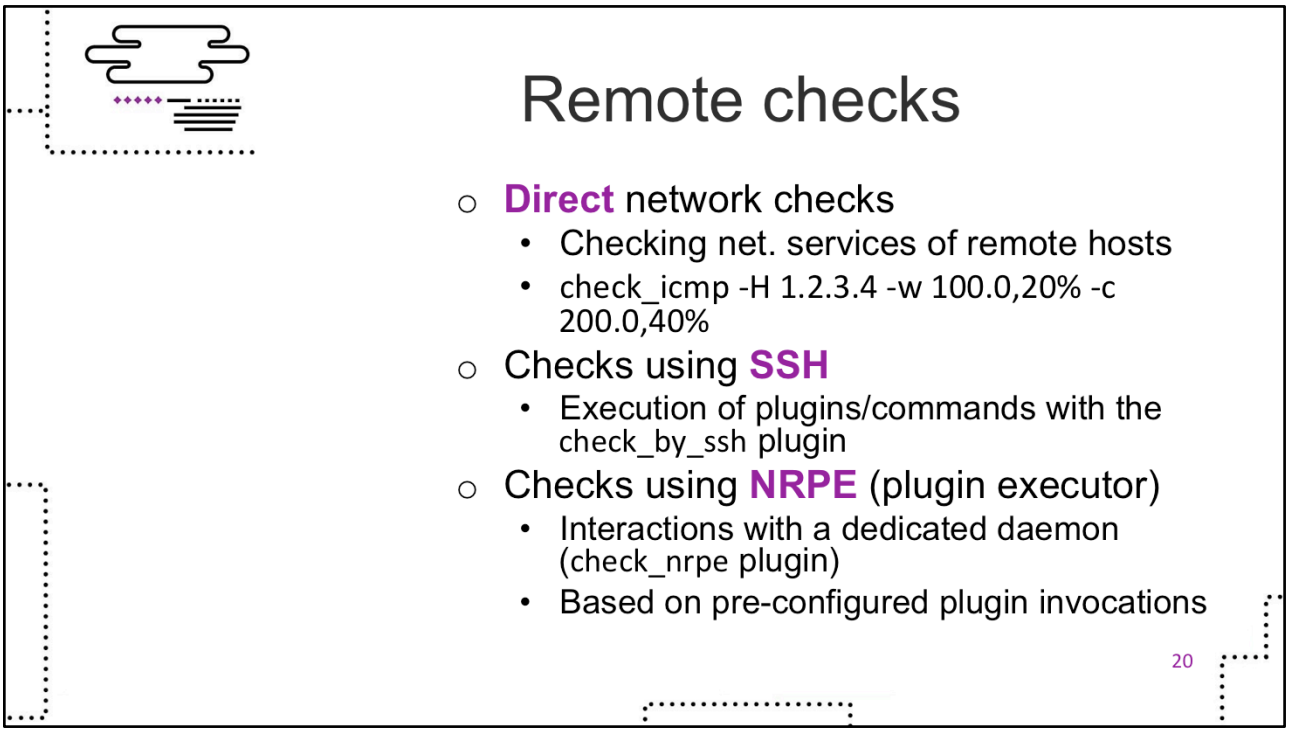


La première catégorie de tests sont les tests locaux. Ces tests permettent d'obtenir des informations localement sur votre système. Tous les tests sont implantés par des plugins. Les tests locaux sont principalement implantés par des plugins qui s'appuient directement sur des commandes système (comme par exemple uptime).

Les plugins peuvent être directement exécutés en ligne de commandes, si vous souhaitez en particulier tester le fonctionnement d'un plugin, et vous assurer qu'il fonctionne correctement avant de l'exécuter avec Nagios. Ici, vous avez trois exemples de tests locaux (ou je devrais dire plus exactement trois exemples de plugins qui implantent des tests locaux).

Le premier est check_disk. Ce plugin va permettre d'évaluer le pourcentage d'espace disponible sur une partition donnée, et ensuite de générer une alerte, si la valeur est supérieure à une valeur seuil. Ici, vous pouvez observer que check_disk prend trois arguments. Le premier qui est -w 30%, qui indique en fait la valeur seuil associée à l'état WARNING. Le deuxième argument -c 15%, qui indique en fait le deuxième seuil (threshold) correspondant à l'état CRITICAL. Et enfin, vous avez un dernier argument -p /var, qui va indiquer la partition qui est utilisée par check_disk pour réaliser son évaluation. Donc check_disk va évaluer l'espace disponible, va générer une alerte avec une alerte d'état WARNING si l'espace disponible est inférieur à 30%, et une alerte indiquant un état CRITICAL si la valeur atteinte est inférieure à 15%.

Les deux plugins suivants sont check_load et check_procs, qui fonctionnent de la même façon. C'est-à-dire que vous spécifiez des valeurs seuils pour définir les états WARNING et CRITICAL. check_load permet d'évaluer la charge sur le système. Comme vous pouvez l'observer, avec check_load, vous avez à chaque fois pour les états WARNING et CRITICAL, trois nombres qui vont indiquer en fait les nombres qui seraient retournés par la commande uptime. Donc, cela correspond à la charge moyenne au cours de la dernière minute, des 5 dernières minutes et des 15 dernières minutes. Check_procs permet d'évaluer le nombre de processus qui fonctionnent sur le système, et ensuite de générer des alertes correspondantes.



Une seconde catégorie de tests sont les tests distants. Nagios fournit différentes façons de réaliser ces tests.

Une première solution, que l'on qualifie de direct network checks, consiste simplement à s'appuyer sur les protocoles réseau pour pouvoir détecter si les services fonctionnent correctement. Ces tests sont directement exécutés depuis le serveur Nagios. Ici, nous avons un exemple d'un tel test, qui s'appelle check_icmp. check_icmp va simplement évaluer l'état d'une machine hôte en utilisant des pings. Il prend trois arguments, le premier correspondant à l'adresse IP de la machine hôte, et les deux autres correspondant aux valeurs seuils pour l'état WARNING et l'état CRITICAL. Vous pouvez constater qu'on a ici deux nombres pour chacune des options : -w et -c. Ces deux nombres correspondent en fait au délai d'aller-retour et au pourcentage de paquets perdus.

Une autre solution pour réaliser des tests distants est d'exécuter les tests en utilisant ssh. Dans ce cas, Nagios va utiliser un plugin dédié, qui s'appelle check_by_ssh. Celui-ci prend en argument le nom d'un plugin suivi d'une liste d'arguments. Le plugin doit tout d'abord être installé sur la machine distante, et va ensuite être exécuté par le plugin check_by_ssh. Donc, vous allez pouvoir exécuter tout type de plugins sur la machine distante, ou tout type de commandes, pourvu que vous ayez les droits pour les exécuter.

Une troisième solution est d'exécuter les tests, en utilisant NRPE. NRPE signifie Nagios Remote Plugin Executor. Dans cette configuration, Nagios va exécuter les tests en utilisant le plugin check_nrpe. Ce plugin va prendre en argument une référence à un plugin, celui que vous souhaitez exécuter sur la machine distante. A partir de ce plugin, Nagios va être capable d'interagir avec NRPE qui fonctionne sur la machine distante et de lancer le plugin pour l'évaluation du test. Dans cette configuration, on est dans une solution où il n'est possible de fournir à Nagios qu'une référence aux plugins que vous souhaitez exécuter (les plugins sont pré-configurés sur la machine distante).

Donc, vous avez deux options, soit SSH pour lequel vous allez pouvoir contrôler complètement la liste des arguments que vous allez exécutés, lorsque vous souhaiterez lancer le test. Et la solution avec NRPE où vous devrez préconfigurer les plugins (à lancer) sur la machine distante. Ensuite, sur le serveur Nagios, vous indiquerez simplement la référence à ce(s) plugin(s) préconfiguré(s). Il existe également d'autres alternatives pour les tests distants. Vous pouvez, par exemple, utiliser SNMP. Vous avez pour cela un plugin check_snmp qui va permettre d'interagir avec des agents, et de collecter des informations de gestion. Dans ce cas, il vous suffira d'associer à vos réponses SNMP des valeurs seuils pour les états WARNING et CRITICAL. Vous avez enfin un autre service qui est disponible : NSCA, qui fait partie également de Nagios et signifie Nagios Service Check Acceptor. Il va vous permettre de démarrer directement des tests depuis les machines distantes, et de simplement rapporter des alertes (push), un peu à la façon des traps SNMP.