



- Pour Commencer
- ▶ Week 0: Introduction to Network and Service Management
- ▶ Week 1: Key Concepts with **SNMP**
- ▼ Week 2: Monitoring with Nagios

Overview of the Content

Lecture 1: Key Concepts and **Architectures** Lesson\_Quiz

Lecture 2: Services, States and Checks (A) Lesson Quiz

Lecture 3: **Configuration and Definitions** Lesson\_Quiz

Lecture 4: Local and Remote Checks Lesson\_Quiz

Lecture 5: Advanced Configurations Lesson\_Quiz

**Practical Exercise** 1: Nagios Installation and **Initial Test** Practical\_Exercise\_Qu

## PRACTICAL EXERCISE 3 (W2 PE3): CONFIGURING POLLING INTERVALS

This last practical exercise aims at experimenting the configuration of polling intervals with Nagios.

It has to be noted that Nagios exhibits a significant delay between the event and its actual depiction on the user interface. This is due to the polling interval e.g. the period of time between two checks performed by the Nagios software.

Fortunately, this polling interval is configurable (as nearly everything in Nagios). For instance, we can define a new service template to specify the polling interval, say, one minute; and inherit all our services from this new template.

First, edit the templates.cfg file and create a new lab-service template with a normal\_check\_interval attribute of 1 minute:

```
define service {
                                           lab-service
        name
                                           local-service
        use
        normal check interval
        register
                                           0
}
```

Next, go back to the nagios-lab.cfg file and arrange it so that all our services inherit from lab-service instead of local-service:

```
define service {
        use
                                          lab-service
; Name of service template to use
        hostgroup name
                                          lab-machines
        service description
                                          PING
        check command
check ping!100.0,20%!500.0,60%
}
```

As usual, a pre-flight check and a restart are requested to show the effects of this modification:



Practical Exercise Qu

# **Practical Exercise** 3: Configuring **Polling Intervals**

Practical Exercise Qu

#### **Evaluations**

Week\_Evaluation Echéance le avril 10, 2022 at 22:00 UTC

Aidez-nous à améliorer ce MOOC

- Week 3: Instrumentation with JMX
- Week 4: Next-Generation Management **Protocols**
- Votre avis nous intéresse

If no errors were detected, the nagios server can be restarted:

# systemctl restart nagios

Bilan de l'exercice : nous avons vu ici comment modifier le paramétrage du monitoring réalisé par Nagios. Plus spécifiquement, nous avons modifié l'intervalle de temps entre deux tests (normal check interval) en utilisant un template. Pour ceux qui avancent vite, vous pouvez considérer d'autres paramètres, comme retry\_check\_interval ou max\_check\_attempts, et observer le comportement de Nagios lorsqu'une machine monitorée n'est plus opérationnelle/joignable (en fonction du nouveau paramétrage).

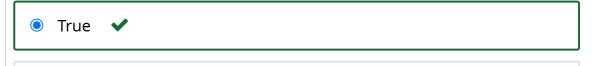
A la fin de cet exercice, n'oubliez pas d'arrêter le service Nagios, avec la commande ci-dessous:

# systemctl stop nagios

### QUESTION W2.PE3.1 (1/1 point)

This exercise demonstrates the extreme flexibility of Nagios. Actually, and given that the monitoring capabilities are all situated in external plugins, the core of Nagios can be considered as an automaton able to organize activities in time.

Which of the following would apply? (NA=1)



False

#### **Correct:**

Sure, and that is why many engineers use Nagios for efficiently monitoring their infrastructure.