# **Weekopdracht 3, opdracht 2 – Monitoring en Logging**

Voor weekopdracht 3 moet er een monitoring en logging server worden opgezet welke een andere server (en de applicaties daarop) in de gaten houdt.

Normaalgesproken moet deze opdracht uitgevoerd worden via Azure virtuele machines, maar aangezien ik hier geen toegang tot heb gekregen doe ik dit via een aantal VirtualBox machines welke een virtueel netwerk delen.

Ik heb in mijn omgeving twee Ubuntu Server 22.04.1 LTS servers; ubuntuserver1 en ubuntuserver2.

*Er draait tevens ook een Windows 10 Pro machine, maar deze is voor weekopdracht 3 niet relevant.*

## Het opzetten van de Munin monitoring omgeving

Als keuze voor monitoring software heb ik Munin gekozen. Munin werkt aan de hand van een master die onderliggende node machines in de gaten houd. Via een webinterface welke gehost is op de master server maakt Munin de resultaten inzichtelijk.

Ik heb ubuntuserver1 uitgekozen als master, wat ubuntuserver2 de node maakt.

### Configuratiestappen Munin-master

Ik heb op ubuntuserver1 de volgende configuratiestappen uitgevoerd:

* Munin en Apache installeren via apt
* In het bestand ‘/etc/munin/munin.conf’ de volgende configuratie toegepast:
  + Boven in het bestand ‘dbdir, htmldir, logdir’ en ‘rundir’ ge-uncomment.
  + Adress 127.0.0.1 (localhost) de alias Master gegeven.
  + Nieuwe node ‘Ubuntu2’ gecreëerd met de volgende instellingen:  
    Graphical user interface, text

    Description automatically generated
* In het bestand ’/etc/munin/apache24.conf’ de volgende configuratie toegepast:
  + Voor de <Directory> ‘/var/ache/munin/www’ en ‘usr/lib/munin/cgi’ heb ik de ‘Require’ instelling op all granted gezet zodat de webserver vanaf elk adres bereikbaar is.
  + Ook heb ik voor deze directories de instellingen ‘Options’ op ‘FollowSymLinks SymLinksIfOwnerMatch’ gezet.
* Ik heb de Munin en Apache service opnieuw laten starten nada ik de ubuntuserver2 configuratie gedaan had.

### Configuratiestappen Munin-node

Ik heb op ubuntuserver2 de volgende configuratiestappen uitgevoerd:

* Munin installeren via apt
* In het bestand ‘/etc/munin/munin-node-conf’ de volgende configuratiestappen uitgevoerd:
  + ‘allow 10.10.10.4’ toegevoegd. Hiermee sta ik toe dat hij connectie maakt met de master.
* Munin-node service opnieuw gestart.

### Het resultaat

Het is nu mogelijk om via een webinterface Ubuntuserver2 te monitoren. Dit webinterface word gehost op ubuntuserver1 en kan bereikt worden via <http://[MACHINE_IP]/munin> -- in mijn geval dus ‘10.10.10.4/munin’.

Chart

Description automatically generated with medium confidence

Het is tevens ook mogelijk om Munin te beveiligen met TLS en een inlogportaal, maar dit viel niet binnen de scope van de opdracht.

## Het opzetten van de logging server,

Voor dit deel van de opdracht moet een van de ubuntuservers logs ontvangen van de andere. Deze logs moeten komen van een Apache en MySQL server. Vervolgens moet de Apache webserver ook beter beveiligd worden met behulp van een “hardening script”.

Als logging tool heb ik Syslog-NG OSE gekozen. Dit is de open source versie van Syslog-NG en heeft meer dan genoeg features voor het doel van deze opdracht.

De hiërarchie blijft hier hetzelfde als het vorige deel van de opdracht; ubuntuserver1 ontvangt de logs van ubuntuserver2.

### Configuratiestappen logging server

Ik heb op ubuntuserver1 de volgende configuratiestappen uitgevoerd:

* Ik heb SyslogNG via apt geinstalleerd.
* In het bestand ‘/etc/syslog-ng/syslog-ng.conf’ een source, destination en log gedefineerd.
  + Aangezien ik de logs ga ontvangen van ubuntuserver2 stel ik in dat ik voor deze source de logs via UDP (en op de default port) op alle interfaces wil gaan ontvangen:

A picture containing text

Description automatically generated

* + Ik moet een destination instellen waar de logs heen gestuurd kunnen worden:

Text

Description automatically generated

* + Tot slot moet ik het daadwerkelijke logging laten plaatsvinden door source(s) aan een destination te koppelen in een ‘log’ functie:

Graphical user interface, text

Description automatically generated with medium confidence

* Om de destination geldig te maken heb ik de destination directory gecreëerd.
* Nadat de configuratie op ubuntuserver2 klaar was heb ik de syslog-ng service opnieuw gestart.

### Configuratiestappen te loggen server

* Ik heb Apache2, MySQL en Syslog-NG via apt geïnstalleerd.
* n het bestand ‘/etc/syslog-ng/syslog-ng.conf’ een source, destination en log gedefineerd.
  + Ik wil logs op dit lokale systeem als een source instellen:

Text

Description automatically generated

* + Vervolgens wil ik ubuntuserver1 als een remote destination instellen. Dit gaat via de default port en UDP:

![Text

Description automatically generated with medium confidence]()

* + Tot slot moet ik het daadwerkelijke logging laten plaatsvinden door source(s) aan een destination te koppelen in een ‘log’ functie:

Text

Description automatically generated

* Ik heb de Syslog-NG service opnieuw gestart.
* Ik heb [dit hardening script](https://raw.githubusercontent.com/ortega-io/apache-hardening/master/http_hard.sh) toegepast om de Apache server beter te beveiligen. Het script doet het volgende:
  + Update Apache tot de nieuwste versie
  + Creëert een back-up van de originele configuratie
  + Maakt fingerprinting moeilijker
  + Zet directory listing uit
  + Zet server-side includes uit
  + Zet symbolic links uit
  + Installeert ‘mod\_security’ en ‘mod\_evasive’ (webapp firewall en D(D)oS/bruteforce bescherming)
  + Limit de request size
  + Beschermt de Apache binary
  + Zet XSS bescherming aan

### Het resultaat

Wanneer er iets toegevoegd word aan de Apache of MySQL logs op ubuntuserver2, dan ziet Syslog-NG dit en worden de logs naar ubuntuserver1 verstuurd en in het ‘/var/log/logserver/log\_10.10.10.5’ bestand neergezet.