

## Inhoud

Starting services.....	7
Listening to Clients.....	7
Basic Services Web Servers .....	8
Database Servers .....	8
Mail Servers .....	8
Serving Local Networks .....	8
Network Resource Servers .....	9
Implementing Security .....	9
Monitoring.....	9
Improving Performance .....	9
Summary .....	9

## Hoofdstuk 2 Introductie tot services (NL)

Wat is een Linux Server?

Een Linux-server draait zonder menselijke interactie en biedt gedeelde bronnen (diensten) aan meerdere gebruikers (clients) in een netwerk. Servers draaien doorgaans zonder grafische interface en gebruiken de commandoregelinterface (CLI) voor beheer.

Diensten starten

Linux-servers kunnen diensten starten op twee manieren:

1. **Daemon:** Een achtergrondproces dat continu draait en wacht op aanvragen van clients.
2. **Super-server:** Een programma dat meerdere diensten beheert en de juiste dienst start op aanvraag van een client. Voorbeelden zijn inetd en de meer geavanceerde xinetd.

Luisteren naar Clients

Servers gebruiken specifieke netwerktechnologieën en protocollen om met clients te communiceren. Dit gebeurt via well-known ports (gestandaardiseerde poorten) zoals:

- **20/21:** FTP (File Transfer Protocol voor bestandsoverdracht)
- **22:** SSH (Secure Shell voor versleutelde toegang)
- **25:** SMTP (Simple Mail Transfer Protocol voor e-mailverzending)
- **53:** DNS (Domain Name System voor naamvertaling)
- **67:** DHCP (Dynamic Host Configuration Protocol voor automatische IP-toewijzing)
- **80:** HTTP (Hypertext Transfer Protocol voor webverkeer)
- **109/110:** POP (Post Office Protocol voor het lezen van e-mail)
- **137-139:** SMB (Server Message Block voor bestand- en printdeling)
- **143/220:** IMAP (Internet Message Access Protocol voor geavanceerde mailboxdiensten)
- **389:** LDAP (Lightweight Directory Access Protocol voor directorydiensten)
- **443:** HTTPS (Hypertext Transfer Protocol Secure voor versleutelde webcommunicatie)
- **2049:** NFS (Network File System voor bestandsharing tussen Unix/Linux systemen)

Basisdiensten Web

Servers

De meest populaire toepassing van Linux-servers is als webserver. Enkele bekende webserver zijn:

- **The Apache Server:** Een modulaire webserver die administratoren in staat stelt om specifieke modules in te schakelen, wat het geheugenverbruik minimaliseert.
- **The nginx Server:** Een moderne webserver die bekendstaat om zijn prestaties en functies zoals load balancing en caching.
- **The Lighttpd Package:** Een lichte webserver die ideaal is voor kleinere toepassingen door zijn lage geheugen- en CPU-gebruik.

## Database Servers

Databases zijn cruciaal voor de meeste toepassingen. Drie populaire open-source databaseservers zijn:

- **The PostgreSQL Server:** Bekend om zijn geavanceerde functies en ACID-naleving, ideaal voor complexe databasebehoeften.
- **The MySQL Server:** Een snelle database met een focus op prestaties, gebruikt in veel webapplicaties. MariaDB is een fork van MySQL met extra functies.
- **The MongoDB Server:** Een NoSQL-database die gegevens opslaat in onafhankelijke documenten, geschikt voor objectgeoriënteerde applicaties.

## Mail Servers

E-mail blijft een belangrijke dienst. De Linux-mailserver bestaat uit verschillende functies:

- **Mail Transfer Agent (MTA):** Beheert inkomende en uitgaande e-mail. Voorbeelden zijn:
  - **Sendmail:** Zeer veelzijdig, maar complex in configuratie.
  - **Postfix:** Eenvoudiger en modulair.
  - **Exim:** Probeert directe levering te maximaliseren.
- **Mail Delivery Agent (MDA):** Bezorgt berichten aan lokale gebruikers. Voorbeelden zijn:
  - **Binmail:** Eenvoudig en populair.
  - **Procmail:** Biedt flexibele configuratie voor e-mailverwerking.
- **Mail User Agent (MUA):** Interactieve programma's voor eindgebruikers om e-mail te lezen. Voorbeelden zijn Evolution en K-Mail.

## Serving Local Networks

- **File Servers:** Delen bestanden in een netwerk. Twee methoden:
  - **NFS (Network File System):** Voor het delen van mappen tussen Linux-systemen.
  - **Samba:** Voor interactie met Windows-systemen via het SMB-protocol.
- **Print Servers:** CUPS (Common Unix Printing System) wordt gebruikt voor het delen van printers in een netwerk, inclusief ondersteuning voor Windows-printers via Samba.

## Network Resource Servers

Enkele belangrijke netwerkdiensten zijn:

- **DHCP:** Dynamic Host Configuration Protocol, zorgt voor automatische IP-adres toewijzing aan clients. De meest populaire Linux DHCP-server is DHCPd.
- **DNS:** Domain Name System, koppelt IP-adressen aan hostnamen. BIND is de meest gebruikte software voor DNS op Linux.
- **Logging:** Rsyslogd en journald worden gebruikt voor lokale en externe logging van systeem informatie.
- **SNMP:** Simple Network Management Protocol, voor het beheren van netwerkapparaten en servers.
- **NTP:** Network Time Protocol, synchroniseert de tijd op servers en clients binnen een netwerk.

## Implementing Security

- **NIS:** Network Information System, deelt een gemeenschappelijke gebruikersdirectory binnen een netwerk.
- **Kerberos:** Beveiligd authenticatieprotocol dat gebruikmaakt van symmetrische sleutelcryptografie.
- **LDAP:** Lightweight Directory Access Protocol, biedt netwerkauthenticatiediensten.
- **Certificate Authority:** Beheert digitale certificaten voor veilige authenticatie.
- **SSH:** Secure Shell, biedt veilige toegang tot servers via encryptie.
- **VPN:** Virtual Private Network, creëert een veilige tunnel voor externe toegang tot een lokaal netwerk.
- **Proxy Server:** Squid is een populaire proxyserver voor het beheren van webverkeer en caching.

## Monitoring

- **Nagios:** Een monitoringtool die SNMP gebruikt om de prestaties en logs van Linux-servers te volgen.

## Improving Performance

Er zijn drie methoden om de prestaties van applicaties te verbeteren:

1. **Clustering:** Verdeelt applicatiefuncties over meerdere servers.
2. **Load Balancing:** Verdeelt clientverzoeken over meerdere servers om de belasting te optimaliseren.

3. **Containers:** Gebruik Docker of Kubernetes om ontwikkelingsomgevingen te verpakken en te verplaatsen tussen servers.

#### Samenvatting

Linux-servers zijn veelzijdig en bieden een breed scala aan diensten voor zowel lokale als wereldwijde toepassingen. Het begrijpen van de verschillende serverdiensten, hun implementatie en hoe ze veilig en efficiënt kunnen worden beheerd, is cruciaal voor systeembeheerders.

### Examen Essentials

#### Beschrijf de manieren waarop serverprogramma's in Linux kunnen worden gestart:

Serverprogramma's in Linux kunnen voortdurend op de achtergrond draaien als een **daemon**-proces of worden gestart door een **super-server daemon** wanneer een client een verzoek doet. Een daemon is een programma dat automatisch op de achtergrond draait en wacht op verzoeken van clients.

#### Leg uit hoe clients weten hoe ze contact kunnen maken met een serverprogramma:

Serverapplicaties luisteren naar clientverbindingen op bekende poorten. Clients moeten een verbindingsverzoek sturen naar de server op de goed gedefinieerde poort voor de applicatie waarmee ze willen communiceren. Deze poorten zijn gestandaardiseerd en vaak geconfigureerd om specifieke serverdiensten aan te duiden.

#### Beschrijf de componenten die doorgaans worden gebruikt in een LAMP-stack:

De **LAMP-stack** maakt gebruik van het **Linux** besturingssysteem, de **Apache** webserver, de **MySQL** databaseserver en de **PHP** programmeertaal om een platform voor webapplicaties te bieden. Deze combinatie van software biedt de basis voor veel dynamische websites en webapplicaties.

#### Beschrijf het verschil tussen een relationele database en een NoSQL-database:

Een **relationele database** slaat gegevensrecords op in afzonderlijke datatabellen. Elke datatype bestaat uit een of meer datavelden die individuele gegevens bevatten. Een gegevensrecord is een instantie van gegevens voor elk dataveld in een tabel. In tegenstelling tot relationele databases, slaat een **NoSQL-database** gegevenswaarden op in documenten. Elk document is onafhankelijk van de andere documenten in de database en kan verschillende gegevenselementen bevatten.

#### Begrijp de manieren waarop een Linux-server bestanden kan delen in een lokaal netwerk:

Linux-servers kunnen het **nfs-utils**-pakket gebruiken om met andere Linux-servers te communiceren en mappen te delen via **NFS** (Network File System). De lokale Linux-server kan mappen van de externe Linux-server koppelen (mounten) alsof het lokale schijven zijn. Linux-servers kunnen ook het **Samba**-pakket gebruiken om bestanden te delen op Windows-lokale netwerken met Windows-clients en -servers, evenals om mappen van Windows-servers te koppelen.

**Begrijp welke serverpakketten meestal worden gebruikt om netwerkfunctionaliteiten op een lokaal netwerk te ondersteunen:**

Het **DHCPd**-pakket biedt DHCP-serverdiensten voor het toewijzen van IP-adressen aan clients. Het **BIND**-pakket biedt DNS-serverdiensten voor zowel clients als servers op een lokaal netwerk voor hostnaamresolutie. Het **net-snmp**-pakket maakt het mogelijk om remote apparaatbeheer uit te voeren via SNMP. Je kunt het **ntpd**-pakket gebruiken om een **NTP** (Network Time Protocol)-tijdserver voor het lokale netwerk te maken.

**Beschrijf hoe je een netwerkdirectoryserver kunt maken met Linux:**

Het **OpenLDAP**-pakket maakt het mogelijk om een **LDAP** (Lightweight Directory Access Protocol)-directory van gebruikers en apparaten op het lokale netwerk te maken. Clients en andere servers kunnen de LDAP-directory gebruiken om gebruikers en apparaten op het netwerk te authenticeren.

**Leg uit hoe je de prestaties van een netwerkapplicatie kunt verbeteren:**

Voor netwerkapplicaties in een omgeving met veel verkeer kun je de prestaties verbeteren door het implementeren van een **cluster** of een **load balancing**-omgeving. In een **cluster** kun je applicatiefuncties verdelen over meerdere servers door een clusterpakket zoals **Apache Hadoop** te gebruiken. Bij **load balancing** kun je clientverbindingen verdelen over meerdere servers met behulp van pakketten zoals **HAProxy** en **Linux Virtual Server (LVS)**.

## Chapter 2 Introduction to services (EN)

What is a Linux Server?

A Linux server runs without human interaction and provides shared resources (services) to multiple users (clients) on a network. Servers typically run without a graphical interface and use the command-line interface (CLI) for management.

Starting services

Linux servers can start services in two ways:

- **Daemon:** A background process that runs continuously and waits for requests from clients.
- **Super-server:** A program that manages multiple services and starts the appropriate service upon client request. Examples include inetd and the more advanced xinetd.

Listening to Clients

Servers use specific network technologies and protocols to communicate with clients. This happens through well-known ports (standardized ports) such as:

- 20/21: FTP (File Transfer Protocol for file transfer)
- 22: SSH (Secure Shell for encrypted access)
- 25: SMTP (Simple Mail Transfer Protocol for email sending)
- 53: DNS (Domain Name System for name translation)
- 67: DHCP (Dynamic Host Configuration Protocol for automatic IP assignment)
- 80: HTTP (Hypertext Transfer Protocol for web traffic)
- 109/110: POP (Post Office Protocol for reading email)
- 137-139: SMB (Server Message Block for file and print sharing)
- 143/220: IMAP (Internet Message Access Protocol for advanced mailbox services)
- 389: LDAP (Lightweight Directory Access Protocol for directory services)
- 443: HTTPS (Hypertext Transfer Protocol Secure for encrypted web communication)
- 2049: NFS (Network File System for file sharing between Unix/Linux systems)

## Basic Services Web Servers

The most popular application of Linux servers is as a web server. Some well-known web servers are:

- **The Apache Server:** A modular web server that allows administrators to enable specific modules, which minimizes memory usage.
- **The nginx Server:** A modern web server known for its performance and features such as load balancing and caching.
- **The Lighttpd Package:** A lightweight web server that is ideal for smaller applications due to its low memory and CPU usage.

## Database Servers

Databases are crucial for most applications. Three popular open-source database servers are:

- **The PostgreSQL Server:** Known for its advanced features and ACID compliance, ideal for complex database needs.
- **The MySQL Server:** A fast database with a focus on performance, used in many web applications. MariaDB is a fork of MySQL with extra features.
- **The MongoDB Server:** A NoSQL database that stores data in independent documents, suitable for object-oriented applications.

## Mail Servers

Email remains an important service. The Linux mail server consists of various functions:

- **Mail Transfer Agent (MTA):** Manages incoming and outgoing email. Examples include:
  - Sendmail: Very versatile, but complex in configuration.
  - Postfix: Simpler and modular.
  - Exim: Tries to maximize direct delivery.
- **Mail Delivery Agent (MDA):** Delivers messages to local users. Examples include:
  - Binmail: Simple and popular.
  - Procmail: Offers flexible configuration for email processing.
- **Mail User Agent (MUA):** Interactive programs for end users to read email. Examples include Evolution and K-Mail.

## Serving Local Networks

- **File Servers:** Share files on a network. Two methods:
  - NFS (Network File System): For sharing folders between Linux systems.
  - Samba: For interaction with Windows systems via the SMB protocol.
- **Print Servers:** CUPS (Common Unix Printing System) is used to share printers on a network, including support for Windows printers via Samba.



## Network Resource Servers

Some important network services are:

- **DHCP:** Dynamic Host Configuration Protocol, provides automatic IP address assignment to clients. The most popular Linux DHCP server is DHCPd.
- **DNS:** Domain Name System, links IP addresses to hostnames. BIND is the most used software for DNS on Linux.
- **Logging:** Rsyslogd and journald are used for local and external logging of system information.
- **SNMP:** Simple Network Management Protocol, for managing network devices and servers.
- **NTP:** Network Time Protocol, synchronizes the time on servers and clients within a network.

## Implementing Security

- **NIS:** Network Information System, shares a common user directory within a network.
- **Kerberos:** Secure authentication protocol that uses symmetric key cryptography.
- **LDAP:** Lightweight Directory Access Protocol, provides network authentication services.
- **Certificate Authority:** Manages digital certificates for secure authentication.
- **SSH:** Secure Shell, provides secure access to servers via encryption.
- **VPN:** Virtual Private Network, creates a secure tunnel for external access to a local network.
- **Proxy Server:** Squid is a popular proxy server for managing web traffic and caching.

## Monitoring

- **Nagios:** A monitoring tool that uses SNMP to track the performance and logs of Linux servers.

## Improving Performance

There are three methods to improve the performance of applications:

- **Clustering:** Distributes application functions across multiple servers.
- **Load Balancing:** Distributes client requests across multiple servers to optimize the load.
- **Containers:** Use Docker or Kubernetes to package development environments and move them between servers.

## Summary

Linux servers are versatile and offer a wide range of services for both local and global applications. Understanding the different server services, their implementation and how they can be managed securely and efficiently is crucial for system administrators.

## **Exam Essentials**

### **Describe the ways to start server programs in Linux:**

Server programs in Linux can either run continually in the background as a **daemon** process or be started from a **super-server daemon** when requested by a client. A daemon is a program that runs automatically in the background and waits for client requests.

### **Explain how clients know how to contact a server program:**

Server applications listen for client connections on well-known ports. Clients must send a connection request to the server on the well-known port for the application they want to interact with. These ports are standardized and often configured to indicate specific server services.

### **Describe the components commonly used in a LAMP stack:**

The **LAMP stack** uses the **Linux** operating system, the **Apache** web server, the **MySQL** database server, and the **PHP** programming language to provide a platform for web applications. This combination of software forms the foundation for many dynamic websites and web applications.

### **Describe the difference between a relational database and a NoSQL database:**

A **relational database** stores data records in individual data tables. Each data type consists of one or more data fields that contain individual data elements. A data record is an instance of data for each data field in a table. In contrast, a **NoSQL database** stores data values in documents. Each document is independent of all other documents in the database and may contain different data elements.

### **Understand the ways a Linux server can share files in a local network:**

Linux servers can use the **nfs-utils** package to communicate with other Linux servers to share folders using **NFS** (Network File System). The local Linux server can mount folders from the remote Linux server as if they were local disks. Linux servers can also use the **Samba** package to share files on Windows local networks with Windows clients and servers as well as map folders located on Windows servers.

### **Understand which server packages are commonly used to support network features on a local network:**

The **DHCPd** package provides DHCP server services to assign IP addresses to clients. The **BIND** package provides DNS server services to both clients and servers on a local network for hostname resolution. The **net-snmp** package allows you to implement remote device management using SNMP. You can use the **ntpd** package to create an **NTP** (Network Time Protocol) time server for the local network.

### **Describe how to create a network directory server using Linux:**

The **OpenLDAP** package allows you to create an **LDAP** (Lightweight Directory Access Protocol) directory of users and devices on the local network. Clients and other servers can use the LDAP directory to authenticate users and devices on the network.

**Explain how to improve the performance of a network application:**

For network applications in a high-traffic environment, you can improve performance by implementing either a **cluster** or a **load balancing** environment. In a **cluster**, you can split application functions between multiple servers by using a cluster package such as **Apache Hadoop**. With **load balancing**, you can distribute client connections between multiple servers using packages such as **HAProxy** and **Linux Virtual Server (LVS)**.