“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №8**

з дисципліни “Операційні системи”:

**Тема: “Збереження службових даних системи та її мережева конфігурація”**

Виконали студенти

групи РПЗ-13А та РПЗ-13Б

Eleven Two Zeroes:

Vlad Sapozhnyk

Max Karpenko

Dmytro Onufriiev

Київ 2024

Робота студентів групи РПЗ-13А та РПЗ-13Б Eleven Two Zeroes: Vlad Sapozhnyk, Max Karpenko and Dmytro Onufriiev.

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими структурами для збереження системних даних - процеси, память, лог-файли та повідомлення про стан ядра.
3. Знайомство зі стандартом FHS.
4. Знайомство з діями при налаштуванні мережі.

**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC.
2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).
3. ОС GNU/Linux (будь-який дистрибутив).
4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки: *Created by Vlad Sapozhnyk***

Glossary of Terms

| **Name of the term** | **Translation** |
| --- | --- |
| implementation | реалізація |
| to store | зберігати |
| process management | управління процесами |
| memory management | управління пам'яттю |
| device drivers | драйвери пристроїв |
| logging | ведення журналу |
| to denote | для позначення |
| rotating | обертовий |
| deprecated | застарілий |

1. **Розкрийте поняття “псевдо файлової системи”, для чого воно потрібно системі?**

A “pseudo-file system” is a virtual file system that does not use physical storage media, such as a hard disk or SSD. Its operation imitates a file system, but the data is stored in RAM or generated dynamically.

“Pseudo-file systems” are used in Linux for various purposes, including:

* Access to system resources: Some “Pseudo-file systems” provide access to system resources, such as network interfaces, processes, or system statistics, in the form of files and directories.
* Information purposes: “pseudo-file system” can be used to provide information about the system or its configuration.
* Temporary data storage: “pseudo-file system” can be used to temporarily store data that does not require permanent storage on physical media.
* Debugging: “pseudo-file system” can be used for software debugging by providing access to internal data or program state.

“Pseudo-file systems” do not have a fixed structure and can be implemented in a variety of ways. Some common “Pseudo-file systems” on Linux include:

/proc: This “pseudo-file system” provides access to information about processes running on the system.

/sys: This “pseudo-file system” provides access to information about the hardware and system configuration.

/dev: This “pseudo-file system” provides access to devices such as hard disks, network interfaces, and ports.

/tmp: This is the path used to temporarily store data.

“Pseudo-file systems” are a valuable tool for Linux administration and development. They provide a simple and convenient way to access system resources, information, and data.

1. **Чому користувачі не так часто звертаються на пряму до каталогу /proc, яким чином з нього можна отримати інформацію?**

Users rarely access the /proc directory directly for several reasons:

* Complexity: The structure of the /proc directory can be complex and confusing for users who are not familiar with its internal organization.
* Inconvenience: Navigating through /proc manually with cd and ls can be awkward and time-consuming, especially when searching for specific information.
* Command line tools: There are more convenient and easier to use command line tools that are specifically designed to access information from /proc. These tools, such as ps, top, and uptime, provide well-structured and understandable output, making them more user-friendly for most users.
* Graphical interfaces: Many Linux distributions offer graphical user interfaces (GUIs) to access information from /proc. These tools, such as System Monitor and htop, visualize information from /proc in a convenient and easy-to-understand format, making them accessible to users who are not familiar with the command line.
* Abstraction:Most Linux administration programs and tools abstract access to /proc, hiding its complexity from users.These tools automatically interpret the data from /proc and present it in a more understandable and convenient format.

Getting information from /proc:

While directly accessing /proc can be tricky, there are several ways to get information from it:

* Command line: You can use various commands such as ps, top, uptime, cat, and grep to access specific data from /proc.
* Scripts:You can write your own scripts in a programming language such as Python or Bash to automate access and process data from /proc.
* GUI tools: You can use graphical tools such as System Monitor or htop to visualize and explore information from /proc.
* Third-party tools:There are many third-party tools that provide convenient access to information from /proc.These tools may have additional features and capabilities compared to the basic command line or GUI tools.

1. **\*Яке призначення файлів /proc/cmdline, /proc/meminfo та /proc/modules?**

The /proc/cmdline, /proc/meminfo, and /proc/modules files on Linux:

1. /proc/cmdline:

This file contains the command line that was used to boot the Linux kernel. It may contain boot parameters passed by the bootloader, such as the kernel name, kernel module options, and other configuration options.

Purpose:

Verify the boot parameters used at system startup.

Troubleshoot problems with kernel loading.

Analyze the kernel configuration.

2. /proc/meminfo:

This file contains detailed information about memory usage in the system. It shows the total amount of memory available, free memory, memory in use, cached memory, virtual memory, and other data related to memory usage.

Purpose:

Monitoring of memory usage in the system.

Identify memory problems, such as memory leaks or memory shortages.

Optimize memory usage in the system.

3. /proc/modules:

This file contains a list of kernel modules that are loaded on the system. It shows the name of each module, version, size, date loaded, and other details.

Purpose:

To check the loaded kernel modules.

Troubleshoot problems with kernel modules.

Remove or load kernel modules.

1. **\*Яке призначення команди free?**

The free command in Linux is used to display information about memory usage in the system.

It provides data on:

Total memory: The total amount of available memory on the system.

Memory in use: The amount of memory used by active processes and cache.

Free memory: The amount of available memory that can be used for new processes or caching.

Buffers: The amount of memory allocated for kernel buffers.

Cached memory: The amount of memory used to cache data and files.

Memory in use: The amount of memory used for memory sharing between processes.

Shared memory: The amount of memory allocated for memory sharing between processes.

Additional options:

-h: Displays information in a more readable format using prefixes (KiB, MiB, GiB, etc.).

-t: Displays memory usage information for each type of memory (e.g., anonymous, cached, buffers).

-m: Displays memory usage information in megabytes (instead of the default kilobytes).

-s: Displays general memory usage statistics only (instead of detailed information).

1. **\*Для чого потрібні лог-файли, наведіть приклади їх застосування?**

Log files, also known as log files, are text files that record events and messages that occur in a system or program. They are used for:

Tracking activity: Log files record user actions, system events, errors, and other informative data that helps administrators and developers track system behavior and identify problems.

Debugging: Analyzing log files can help in identifying and fixing errors, crashes, and other problems in a system or application.

Auditing: Log files can be used to record information about system access, configuration changes, and other activities, which can be useful for auditing and regulatory compliance.

Performance analysis: Log files can contain information about response times, resource utilization, and other data that can be used to analyze the performance of a system or application.

Diagnostics: Log files can contain information about network activity, database errors, and other events that can help diagnose problems with the network, databases, and other system components.

Examples of log file applications:

Web servers: Web servers record log files that contain information about website requests, HTTP errors, user IP addresses, and other data.

System logs: Operating systems write system logs that contain information about system boot, system events, kernel errors, and other data.

Program logs: Software applications often write log files that contain information about program usage, program errors, and other data.

Security logs: Security systems write log files that contain information about authorization attempts, security breaches, and other security-related events.

Network logs: Network devices write log files that contain information about network traffic, network errors, and other network-related data.

1. **\*\*Яке призначення файлу /var/log/dmesg?**

The /var/log/dmesg file on Linux is used to store the output of the dmesg command. This command displays Linux kernel messages that are generated while the system is booting and running.

1. **\*\*Для чого розроблено FHS?**

The Linux File System Hierarchy (FHS) is a standardized directory structure for organizing files and data in Linux operating systems. It is designed to provide consistency and predictability in the location of files across different Linux distributions.

1. **\*\*Які основні команди є у Linux для перегляду та конфігурації мережі**

View network information:

ifconfig (or ip a): Displays network interface information such as IP addresses, MAC addresses, subnet masks, and status.

route (or ip r): Displays the routing table used to route traffic to different networks.

ping : Checks the availability of a host on the network by sending ICMP packets to it.

nslookup : Converts hostnames to IP addresses and vice versa.

hostname : Displays or changes the system hostname.

Configure the network:

ifconfig (or ip a): Can be used to configure IP addresses, subnet masks, MAC addresses, and other network interface settings.

route (or ip r): Can be used to add, delete, and modify routes in the routing table.

dhclient : Automatically obtains the IP address, subnet mask, and other network parameters from a DHCP server.

nmtui : A graphical user interface for configuring the network.

netplan : A configuration file to describe the system's network settings.

Other useful commands:

dig : DNS client used to obtain information about DNS records.

tcpdump : A tool for capturing and analyzing network traffic.

mtr : A tool for monitoring packet routing and identifying network problems.

wireshark : A graphical tool for capturing and analyzing network traffic.

**Хід роботи. *Created by Max Karpenko***

* 1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
  2. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse ***(якщо виконуєте ЛР у 401 ауд.)*** та запустіть термінал.
  3. Запустіть віртуальну машину Ubuntu\_PC ***(якщо виконуєте завдання ЛР через академію netacad)***
  4. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.
  5. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials - Lab 13: Where Data is Stored*** та ***Lab 14: Network Configuration.*** Створіть таблицю для опису цих команд

| Назва команди | Її призначення та функціональність |
| --- | --- |
| su | Змінюємо поточного користувача на root |
| ls /proc | Переглядаємо вміст системного каталогу **/proc** (для цього потрібні права доступу root) |
| cat /etc/passwd | Відображення вмісту файлу /etc/passwd, який містить інформацію про користувачів системи |
| ifconfig | Показує конфігурацію мережевих інтерфейсів, використовується для налаштування або перегляду параметрів мережі |
| ip addr | Відображає стан мережевих інтерфейсів, також використовується для налаштування мережі; заміна ifconfig в новіших дистрибутивах |
| ping | Відправляє ICMP ECHO\_REQUEST до мережевих вузлів, використовується для перевірки доступності вузла в мережі |
| netstat | Відображає стан мережевих з'єднань, маршрутизацію, статистику інтерфейсів, маски тощо |
| traceroute | Визначає і відображає маршрут (трасування пакетів) до мережевого вузла |

* 1. Виконайте практичні завдання у терміналі (продемонструйте скріншоти):
* **в даній лабораторній роботі використовувалась команда *cat*, дослідіть її можливості та опишіть для яких задач вона призначена;**

The cat command is one of the most fundamental commands in the Unix and Linux environments, used to display the text of files on the screen, as well as to combine them and redirect them to other files or programs. The main tasks for which the cat command is intended include:

**Viewing the contents of a file:** Using cat to quickly view small or medium-sized text files is very convenient because it allows users to quickly view the contents of a file right in the terminal.

**Combining multiple files:** cat can also be used to merge multiple files into one large file. This is done by sequentially outputting the contents of multiple files and redirecting the result to a new file.

**Append content to the end of a file:** With cat, users can append the contents of one or more files to the end of an existing file using output redirection.

**Creating a new file:** With cat, users can create a new file by typing text from the keyboard and redirecting it to a file.

**Display line numbers:** To better analyse the text of files, cat can display line numbers, making it useful for parsing output in scripts or while programming.

**Detect invisible characters:** The cat command can be configured to display non-transferable and control characters, allowing users to understand the structure of a

file, including line feeds, tabs, and other special characters.

* **\*продемонструйте приклади, коли команда *cat* використовується для створення файлу, перегляду вмісту файлу, перенаправлення інформації у інший файл,** склеювання декількох файлів в один;

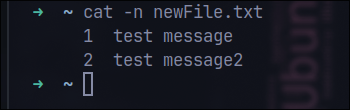
**Reading files:** cat can be used to display the textual content of a file on the screen. This is useful for viewing short files.



**Combining files:** Cat can be used to merge multiple files into a single file. The output from the first file becomes the input to the next file.



**Display line numbers: cat can display line numbers for all output lines, which is especially useful when parsing long files.**



* **\*які параметри команди *cat* треба використати, щоб пронумерувати рядки файлу, відобразити недруковані символи, видалити порожні рядки?**

To number the lines of a file when using the cat command, use the -n option. To display unprintable characters, use the -A option, which shows unprintable characters and line ends. To remove blank lines, you can use the grep command with cat to filter, for example: cat file | grep ., where grep . removes lines containing no characters.

* \*\*опишіть можливості команди *dig* та наведіть приклади;

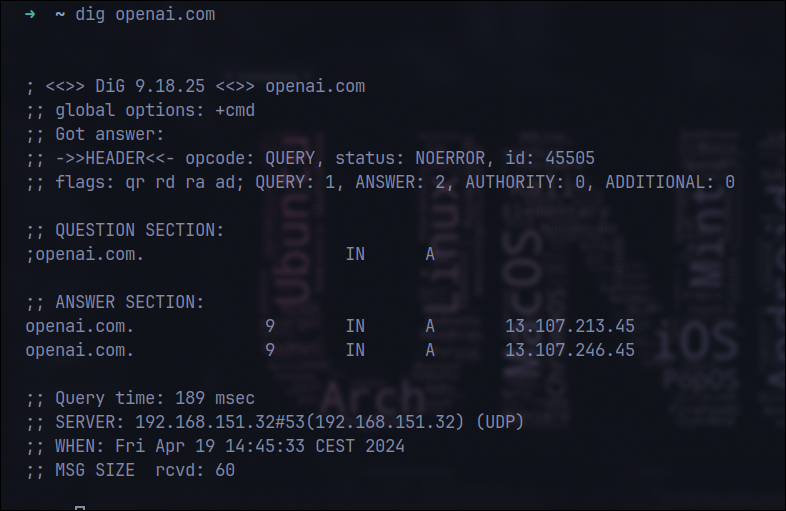
**Query DNS records:** the ability to query specific types of DNS records, such as A (IPv4 address), AAAA (IPv6 address), MX (mail exchanger), TXT (text records), etc.

**Testing DNS servers:** checking the response speed of servers and analysing the responses to diagnose possible DNS issues.

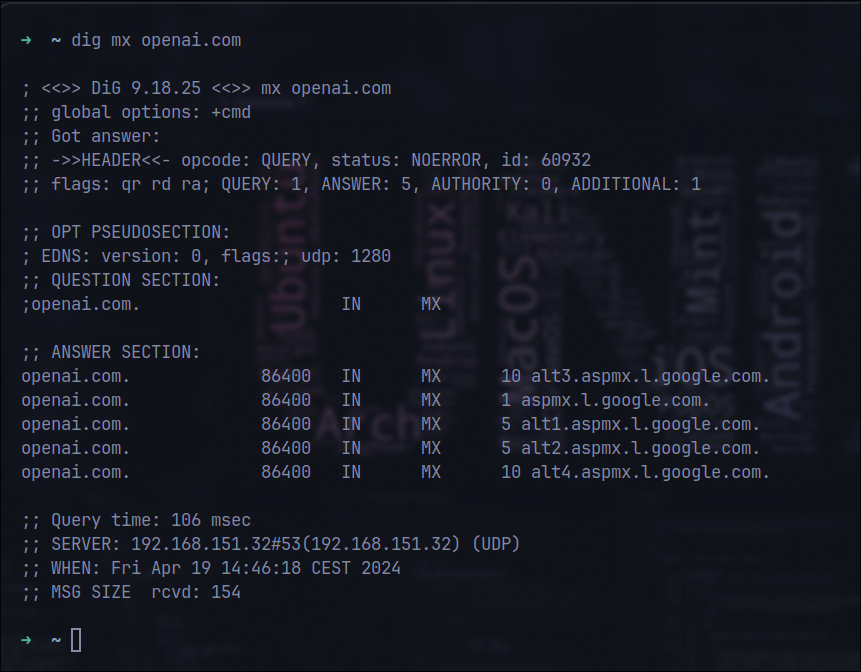
**Displaying detailed information:** dig allows you to display the DNS query and response process in detail, including response time and other statistics.

**Use of specific DNS servers:** the ability to direct queries to specific DNS servers instead of using the default servers.

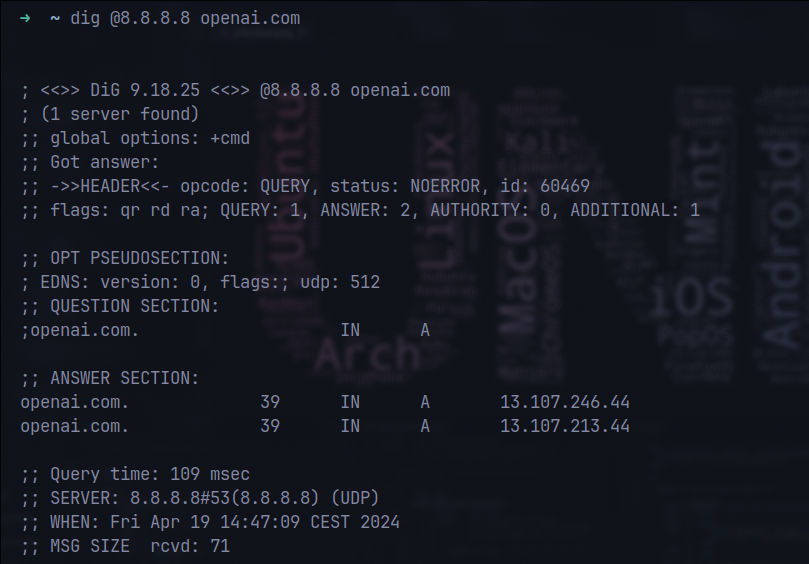
**Requesting a basic A record:**

****

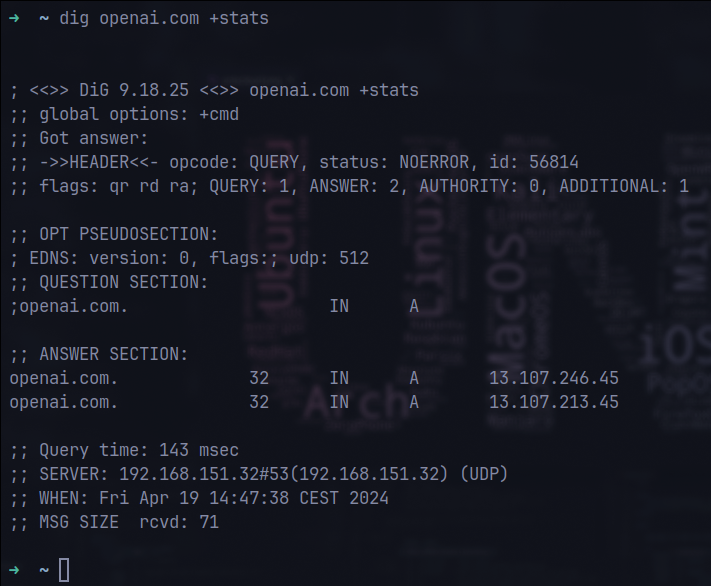
**Requesting MX records:**

****

**Specific DNS server:**

****

**Get full request statistics:**

****

* \*\*опишіть можливості команди *netstat* та наведіть приклади;.

**Display all active connections:** You can view all active connections to and from the system.

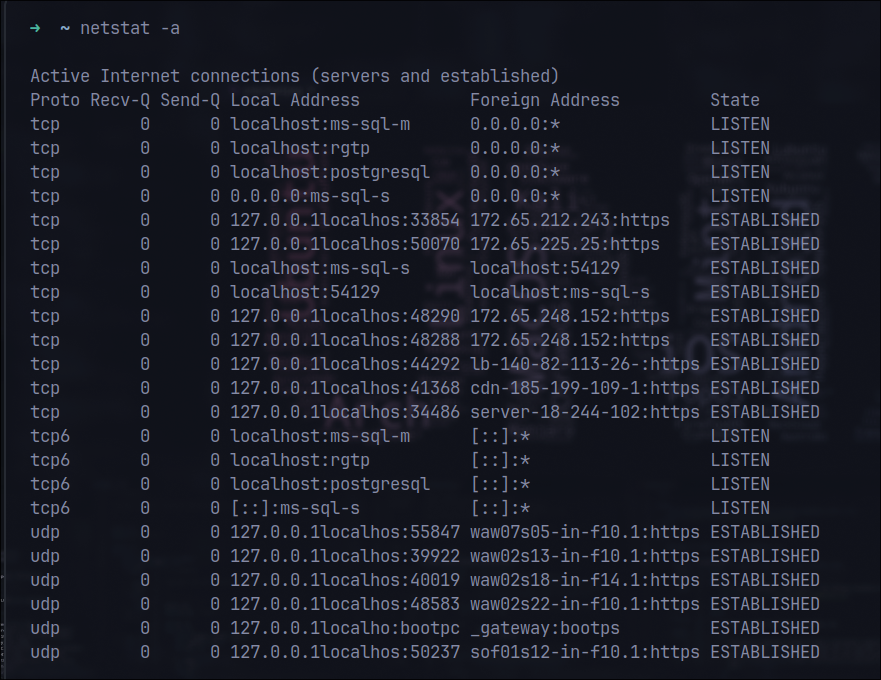
**View protocol statistics:** You can get statistics for TCP, UDP, ICMP, and other protocols.

**Listening ports view:** Identify ports that are open and ready to receive connections.

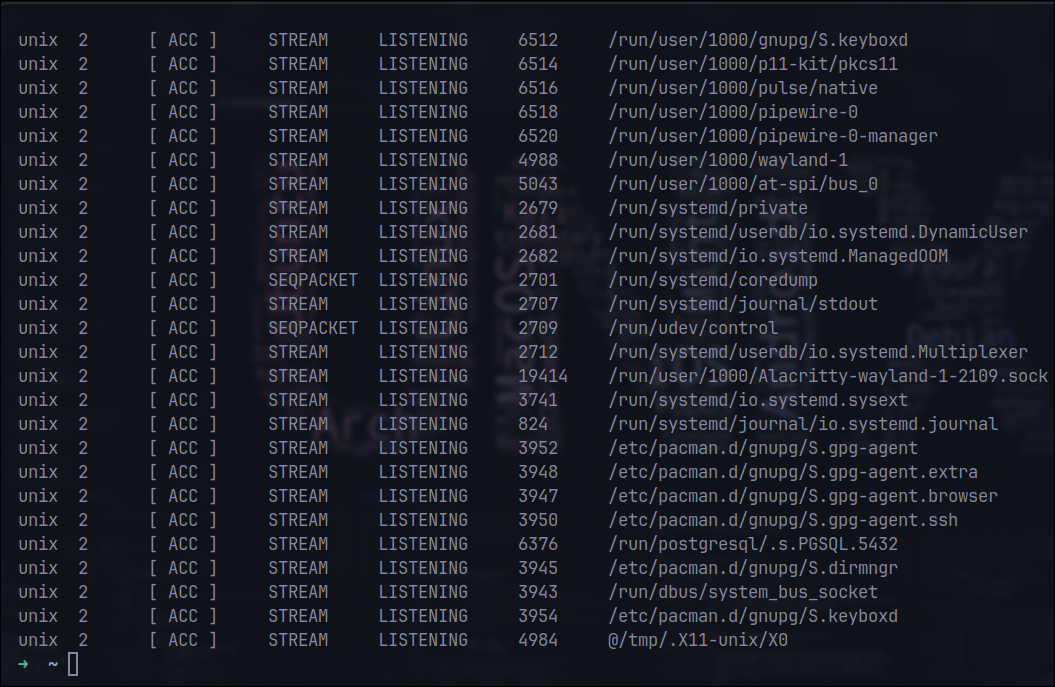
**Monitoring network interfaces:** displays statistics on incoming and outgoing packets for each network interface.

**View routing table:** allows you to view the system's routing table.

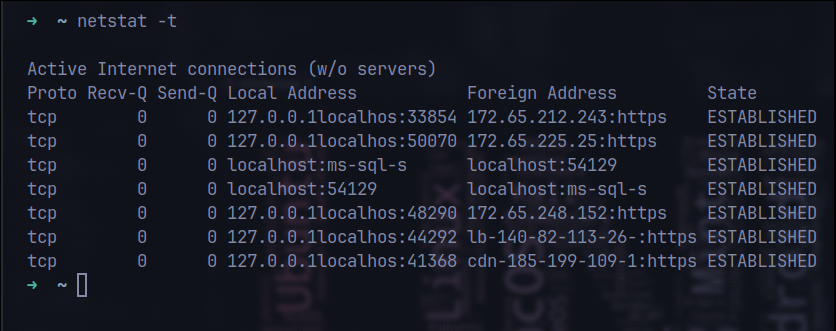
**View all active connections:**

****

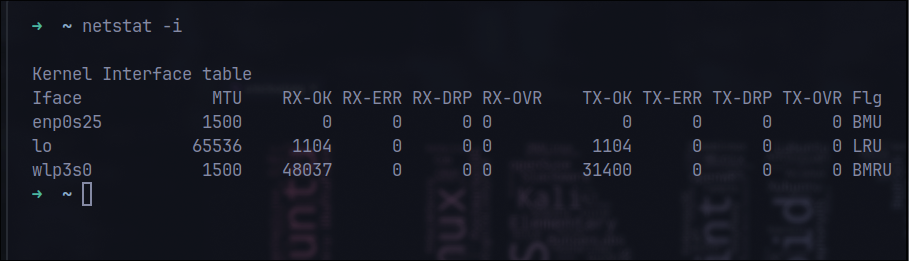
**View the listening ports:**

****

**View TCP connections and port states:**

****

**Statistics on network interfaces:**

****

**Routing table:**

****

**Контрольнi запитання: *Created by Dmytro Onufriiev***

**Висновок:**