

## «РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ СЕТЕВЫХ КОНФИГУРАЦИИ»

Мурат Дильназ Халелкызы

kabae@list.ru

студентка 1 курса образовательной программы «Администрирование систем и сетей»  
Атырауский университет имени Х. Досмухамедова, г. Атырау, Республика Казахстан  
Научный руководитель — сеньор-лектор, магистр Габдулова А.Е.

Современные компьютерные сети являются важной частью функционирования информационных систем, серверных инфраструктур, облачных сервисов и интернет-ресурсов. От стабильности сетевой конфигурации напрямую зависит работа web-сайтов, приложений, баз данных и других цифровых сервисов.

При настройке серверов и сетевого оборудования системные администраторы часто сталкиваются с проблемами неправильной конфигурации сети: отсутствием маршрута по умолчанию, ошибками DNS-разрешения, недоступностью удалённых узлов или закрытыми сетевыми портами. Поиск таких проблем обычно выполняется вручную с использованием различных сетевых команд и утилит. Этот процесс может занимать значительное время и требует высокой квалификации специалиста.

Актуальность темы обусловлена необходимостью автоматизации процессов диагностики сетевых конфигураций серверов. Использование автоматизированных инструментов позволяет значительно ускорить выявление сетевых ошибок, повысить точность проверки и снизить вероятность человеческого фактора при анализе состояния сети.

Цель работы: разработка системы автоматизированного тестирования сетевых конфигураций на базе Linux-сервера с использованием Bash-скрипта.

Задачи исследования:

- развернуть серверную среду Linux в виртуальной машине;
- выполнить настройку сетевых интерфейсов и подключение к сети;
- установить необходимые сетевые диагностические утилиты;
- разработать Bash-скрипт для автоматической проверки параметров сети;
- реализовать проверку маршрутизации, DNS, доступности узлов и сетевых портов;
- сформировать автоматический отчёт о результатах проверки.

Разработанная система позволяет автоматизировать процесс диагностики сетевой инфраструктуры и может использоваться в учебных целях, а также при администрировании серверов.

Сетевая конфигурация — это совокупность параметров и настроек, определяющих работу сетевых интерфейсов устройства и его взаимодействие с другими узлами сети.

К основным параметрам сетевой конфигурации относятся IP-адрес, маска сети, шлюз по умолчанию и DNS-серверы. Корректная настройка сетевой конфигурации является важным условием стабильной работы серверов и сетевых сервисов.

Маршрутизация — это процесс выбора пути передачи сетевых пакетов от источника к получателю через сеть.

Одним из ключевых элементов маршрутизации является маршрут по умолчанию (default route). Он используется в том случае, когда система не знает точного маршрута к удалённому узлу и направляет трафик через основной шлюз.

DNS (Domain Name System) — это система доменных имён, которая преобразует доменные имена сайтов в IP-адреса.

Например, пользователь вводит адрес сайта, а DNS определяет соответствующий IP-адрес сервера. Без корректной работы DNS невозможно подключиться к интернет-ресурсам по доменному имени. Для диагностики DNS используется утилита `dig`, которая позволяет проверить корректность разрешения доменных имён.

Протокол ICMP и команда `ping`. ICMP (Internet Control Message Protocol) — это сетевой протокол, используемый для передачи диагностических сообщений и сообщений об ошибках. На основе этого протокола работает команда `ping`, которая позволяет проверить доступность удалённого узла в сети. Если узел отвечает на `ping`-запрос, значит соединение между устройствами установлено.

HTTPS и утилита `curl`. HTTPS — это защищённая версия протокола HTTP, используемая для передачи данных между клиентом и сервером с использованием шифрования.

Для проверки доступности веб-ресурсов используется утилита `curl`, которая позволяет отправлять HTTP и HTTPS запросы к серверам.

Сетевые порты и утилита `nmap`. Сетевой порт — это логический канал связи, через который сетевые приложения обмениваются данными.

Каждый сервис работает на определённом порту.

Например:

порт 22 — SSH

порт 80 — HTTP

порт 443 — HTTPS

Для проверки открытых портов используется утилита `nmap`, которая позволяет сканировать сетевые порты и определять их состояние.

Автоматизация сетевой диагностики позволяет объединить основные проверки сети в единую систему и выполнять их без ручного ввода команд. В рамках исследования разработан Bash-скрипт, обеспечивающий проверку маршрутизации, работы DNS, доступности узлов, веб-соединения и состояния портов с последующим формированием отчёта о состоянии конфигурации сервера. Использование данного подхода позволяет сократить время диагностики и повысить эффективность администрирования.

В рамках исследования была выполнена настройка серверной среды, установка сетевых инструментов и разработка Bash-скрипта для автоматической проверки сетевой конфигурации.

Для реализации проекта использовалась виртуальная среда Oracle VirtualBox, в которой была развернута виртуальная машина с серверной операционной системой.

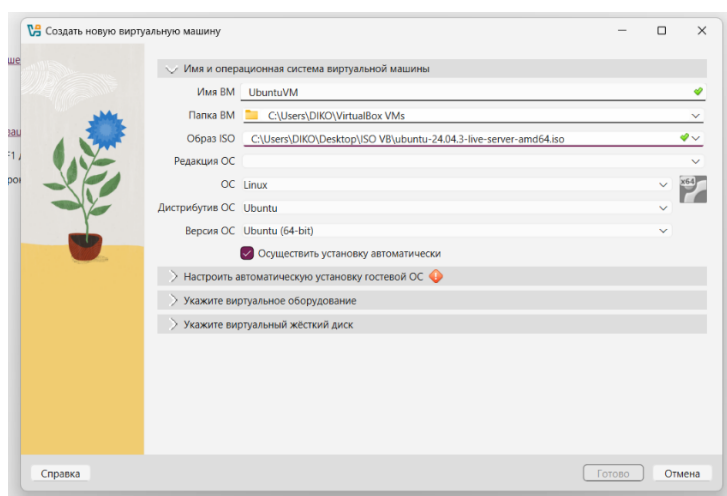


Рисунок 1. Создание виртуальной машины в Oracle VirtualBox

На данном этапе выполнялась настройка аппаратных параметров виртуальной машины. Были заданы объём оперативной памяти и количество виртуальных процессоров, необходимых для стабильной работы серверной системы.

Далее был создан виртуальный жёсткий диск, который используется для хранения файлов операционной системы и данных сервера. Был выбран формат VDI и задан размер виртуального диска.

На следующем этапе был настроен первый сетевой адаптер виртуальной машины в режиме NAT. Такой режим позволяет серверу получать доступ в интернет через сетевое соединение основного компьютера.

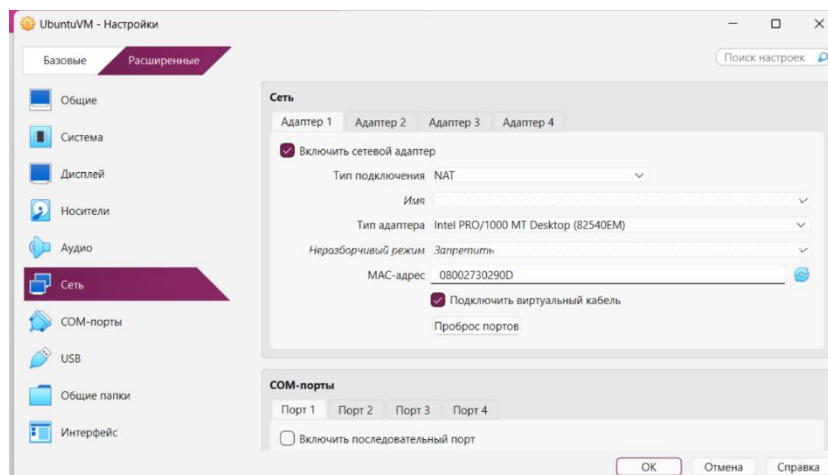


Рисунок 2 — Настройка сетевого адаптера в режиме NAT

Второй сетевой адаптер был настроен как виртуальный адаптер Host-Only. Этот режим позволяет создать отдельную изолированную сеть между виртуальной машиной и хост-компьютером.

В процессе установки Ubuntu Server была выполнена настройка зеркала репозитория, которое используется для загрузки и обновления программных пакетов системы.

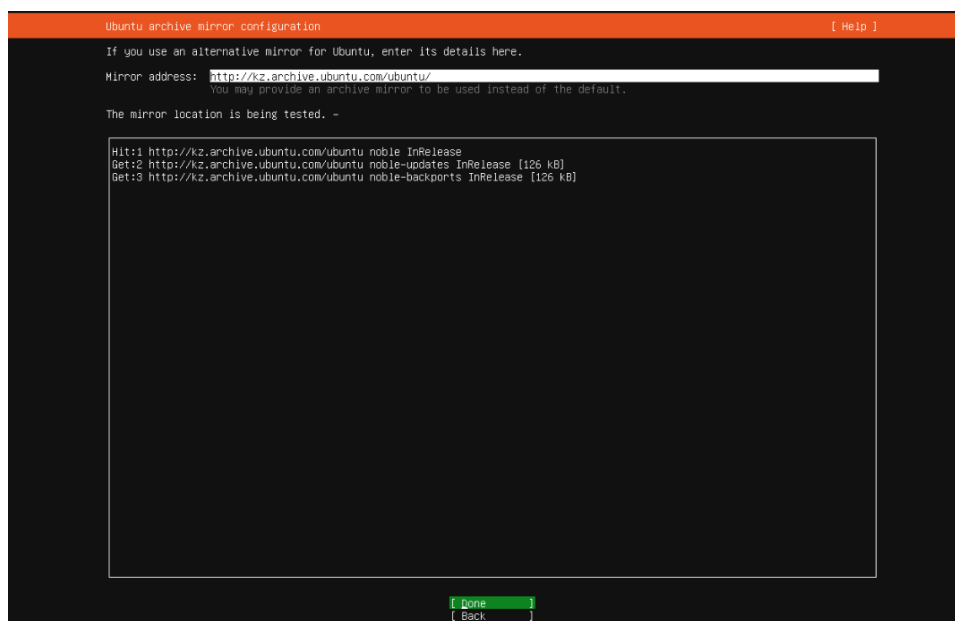


Рисунок 3 — Настройка зеркала репозитория при установке Ubuntu Server

На данном этапе выполнялась настройка параметров пользователя и имени сервера. Был создан пользователь netadmin и задано имя сервера ubuntu-lab.

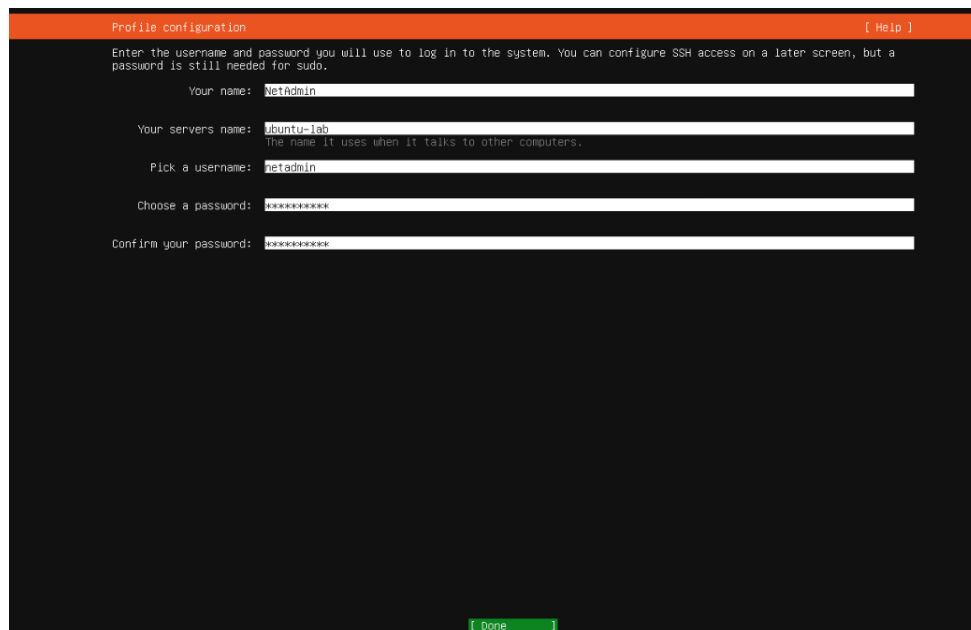


Рисунок 4 — Настройка пользователя и имени сервера

После настройки основных параметров начался процесс установки операционной системы Ubuntu Server и базовых компонентов системы.

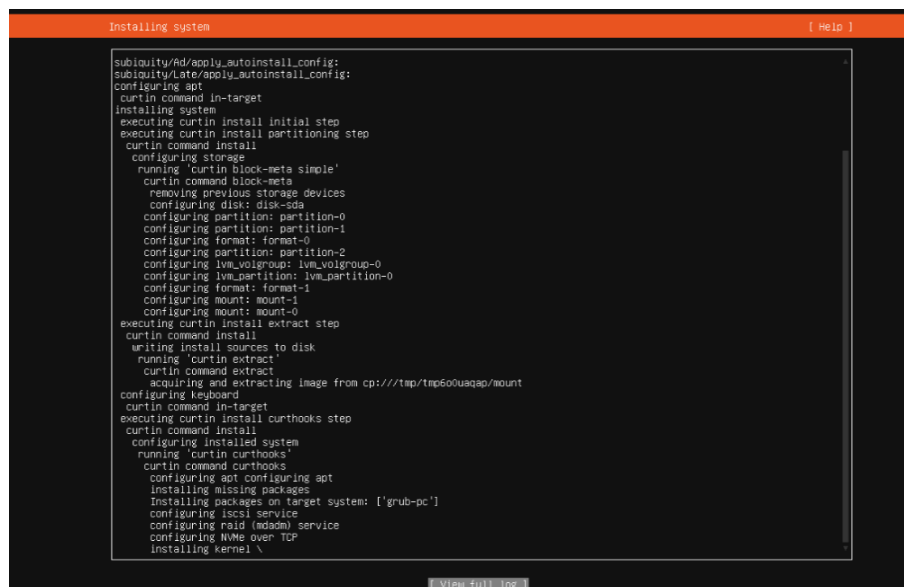


Рисунок 5 — Процесс установки операционной системы Ubuntu Server

После завершения установки был выполнен вход в систему под созданным пользователем netadmin, что подтверждает корректную установку сервера.

Для проверки сетевой конфигурации была использована команда ip a, которая отображает активные сетевые интерфейсы и назначенные IP-адреса.

```

netadmin@ubuntu-lab:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:30:29:0d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86230sec preferred_lft 86230sec
    inet6 fd17:625c:f037:2:a00:27ff:fe30:290d/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86231sec preferred_lft 14231sec
    inet6 fe80::a00:27ff:fe30:290d/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:3f:86:dc brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 metric 100 brd 192.168.56.255 scope global dynamic enp0s8
        valid_lft 430sec preferred_lft 430sec
    inet6 fe80::a00:27ff:fe3f:86dc/64 scope link
        valid_lft forever preferred_lft forever
netadmin@ubuntu-lab:~$

```

Рисунок 6 — Проверка сетевых интерфейсов с помощью команды ip

Далее была выполнена проверка сетевого соединения с помощью команды ping к публичному DNS-серверу 8.8.8.8. Полученные ответы подтверждают наличие доступа в интернет.

```

netadmin@ubuntu-lab:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=255 time=111 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=255 time=136 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=255 time=192 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=255 time=155 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=255 time=111 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=255 time=111 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=255 time=127 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=255 time=111 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=255 time=112 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=255 time=112 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=255 time=139 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=255 time=152 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=255 time=111 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=255 time=112 ms
^C
--- 8.8.8.8 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13098ms
rtt min/avg/max/mdev = 110.507/127.909/191.917/23.655 ms
netadmin@ubuntu-lab:~$

```

Рисунок 7 — Проверка доступности сети с помощью команды ping

Перед установкой дополнительных инструментов была выполнена команда sudo apt update для обновления списка доступных пакетов системы.

```

netadmin@ubuntu-lab:~$ sudo apt update
[sudo] password for netadmin:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://kz.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://kz.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://kz.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
71 packages can be upgraded. Run 'apt list --upgradable' to see them.
netadmin@ubuntu-lab:~$

```

Рисунок 8 — Обновление списка пакетов системы

После обновления системы были установлены необходимые сетевые утилиты: dnsutils, curl, nmap и net-tools, которые используются для диагностики сетевых параметров.

```
Selecting previously unselected package dnstools.
(Reading database ... 84602 files and directories currently installed.)
Preparing to unpack .../0-dnstools_1.9.18-0ubuntu0.24.04.2.all.deb ...
Unpacking dnstools (1:9.18.39-0ubuntu0.24.04.2) ...
Selecting previously unselected package libblas3:amd64.
Preparing to unpack .../1-libblas3_3.12.0-3build1.1_amd64.deb ...
Unpacking libblas3:amd64 (3.12.0-3build1.1) ...
Selecting previously unselected package liblinear4:amd64.
Preparing to unpack .../2-liblinear4_2.3.0+dfsg-5build1_amd64.deb ...
Unpacking liblinear4:amd64 (2.3.0+dfsg-5build1) ...
Selecting previously unselected package liblua5.4-0:amd64.
Preparing to unpack .../3-liblua5.4-0_5.4.6-3build2_amd64.deb ...
Unpacking liblua5.4-0:amd64 (5.4.6-3build2) ...
Selecting previously unselected package libssh2-1t64:amd64.
Preparing to unpack .../4-libssh2-1t64_1.11.0-4.1build2_amd64.deb ...
Unpacking libssh2-1t64:amd64 (1.11.0-4.1build2) ...
Selecting previously unselected package net-tools.
Preparing to unpack .../5-net-tools_2.10-0.1ubuntu4.4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4.4) ...
Selecting previously unselected package nmap-common.
Preparing to unpack .../6-nmap-common_7.94+git20230807.3be01efb1+dfsg-3build2_all.deb ...
Unpacking nmap-common (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Selecting previously unselected package nmap.
Preparing to unpack .../7-nmap_7.94+git20230807.3be01efb1+dfsg-3build2_amd64.deb ...
Unpacking nmap (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Setting up net-tools (2.10-0.1ubuntu4.4) ...
Setting up libblas3:amd64 (3.12.0-3build1.1) ...
update-alternatives: using /usr/lib/x86_64-linux-gnu/blas/libblas.so.3 to provide /usr/lib/x86_64-linux-gnu/libblas.so.3 (libblas mode)
Setting up nmap-common (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Setting up liblua5.4-0:amd64 (5.4.6-3build2) ...
Setting up libssh2-1t64:amd64 (1.11.0-4.1build2) ...
Setting up liblinear4:amd64 (2.3.0+dfsg-5build1) ...
Setting up nmap (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.7) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Рисунок 9 — Установка сетевых диагностических утилит

Для разработки скрипта автоматической диагностики была создана отдельная директория проекта netcheck.

```
netadmin@ubuntu-lab:~$ mkdir ~/netcheck
netadmin@ubuntu-lab:~$ cd ~/netcheck
netadmin@ubuntu-lab:~/netcheck$
```

Рисунок 10 — Создание директории проекта netcheck

На данном этапе был разработан Bash-скрипт netcheck.sh, который выполняет автоматическую проверку сетевых параметров сервера.

```
section "3) Examination ping"
need_cmd ping && {
    if ping -c 3 -W 2 "$TARGET_HOST" >/dev/null 2>&1; then
        ok "Ping passes"
    else
        fail "Ping NO passes"
    fi
}

section "4) Examination DNS"
need_cmd dig && {
    if dig @$DNS_SERVER "$DNS_NAME" +time=2 +tries=1 +short | grep -qE '^0-9';
    ok "DNS works"
    else
        fail "DNS NO works"
    fi
}

section "5) Examination HTTPS"
need_cmd curl && {
    if curl -I -m 6 -s https://example.com | head -n 1 | grep -q "HTTP/"; then
        ok "HTTPS works"
    else
        fail "HTTPS NO works"
    fi
}

section "6) Checking ports"
need_cmd nmap && {
    for p in "${PORTS[@]}; do
        if nmap -p "$p" --open -Pn "$PORT_HOST" 2>/dev/null | grep -q "$p/tcp open";
        ok "Port $p open"
        else
            fail "Port $p closed"
        fi
    done
}

section "7) Result"
info "The log has been saved: $LOG_FILE"

netadmin@ubuntu-lab:~/netcheck$ chmod +x netcheck.sh
netadmin@ubuntu-lab:~/netcheck$
```

Рисунок 11 — Фрагмент кода Bash-скрипта для диагностики сети

После запуска скрипта выводятся результаты автоматической проверки сетевой конфигурации. Все проверки классифицируются по статусам ОК и FAIL и сохраняются в лог-файл.

```
1) Network information
=====
ip                UNKNOWN    127.0.0.1/8   ::1/128
enp0s3           UP         10.0.2.15/24 metric 100 fd17:625c:f037:2:a00:27ff:fe30:290d/64 fe80::a00:27ff:fe30:290d/64
enp0s8           UP         192.168.56.101/24 metric 100 fe80::a00:27ff:fe3f:86dc/64

default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
192.168.1.1 via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.101 metric 100
=====
2) Examination default route
=====
[OK] Default route found
=====
3) Examination ping
=====
[FAIL] No command: ping
=====
4) Examination DNS
=====
netadmin@ubuntu-lab:~/netchecks
```

Рисунок 12 — Результат выполнения скрипта диагностики сети

**Заключение.** В ходе выполнения данной работы была разработана система автоматизированного тестирования сетевых конфигураций на базе Linux-сервера. Для реализации проекта была создана виртуальная серверная среда в Oracle VirtualBox и установлена операционная система Ubuntu Server.

В процессе работы была выполнена настройка сетевых интерфейсов, проверена доступность сети и установлены необходимые диагностические утилиты. На основе этих инструментов был разработан Bash-скрипт, который автоматически выполняет проверку основных параметров сетевой конфигурации.

Разработанный скрипт позволяет проверять корректность маршрутизации, доступность сетевых узлов, работу DNS-системы, HTTPS-соединения и состояние сетевых портов. Результаты проверки автоматически выводятся в понятном виде и классифицируются по статусам OK или FAIL, что упрощает анализ состояния сети.

Использование автоматизированной проверки позволяет значительно сократить время диагностики сетевых проблем, уменьшить вероятность ошибок при ручной проверке и повысить удобство администрирования серверных систем.

Таким образом, разработанная система может использоваться как инструмент для диагностики сетевых конфигураций, а также служить основой для дальнейшего развития систем автоматического мониторинга серверной инфраструктуры.

#### Список использованной литературы:

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – Юбилейное изд. – Санкт-Петербург : Питер, 2020.
2. Shotts, W. E. The Linux Command Line : A Complete Introduction / W. E. Shotts. – 5th Internet ed. – 2019.
3. Тулупова, О. П. Сетевое администрирование Linux : учебно-методическое пособие / О. П. Тулупова, В. Р. Ганиева. – Уфа : УГНТУ, 2020.
4. Справочник по командному процессору Bash / GNU Project ; пер. на рус. яз. – 2018.
5. Ubuntu Server documentation: official documentation / Canonical Ltd. – 2026. – URL: <https://ubuntu.com/server/docs/>
7. Nmap Reference Guide // Nmap Network Scanning. – 2026. – URL: <https://nmap.org/book/man.html>
8. BIND 9 Administrator Reference Manual // ISC BIND 9 Documentation. – 2026. – URL: <https://bind9.readthedocs.io/en/v9.20.21>
9. Oracle VirtualBox User Guide for Release 7.2 // Oracle Documentation. – 2026. – URL: <https://docs.oracle.com/en/virtualization/virtualbox/7.2/user/EN-VBOX-7-2-USER.pdf>
10. curl man page // curl Documentation. – 2026. – URL: <https://curl.se/docs/manpage.html>
11. ip-route(8) – routing table management // Ubuntu Manpages. – 2026. – URL: <https://manpages.ubuntu.com/manpages/noble/man8/ip-route.8.html>