

# Работа с DNS

## Задачи

- 1. Установить и настроить авторитетного DNS сервера
- 2. Настроить DNS для работы в соответствии со стандартом RFC 2136
- 3. Установка и настройка рекурсивного DNS
- 4. Выпустить сертификат безопасности
- 5. Проверить сертификата безопасности

## Построение стенда

Схема виртуального лабораторного стенда

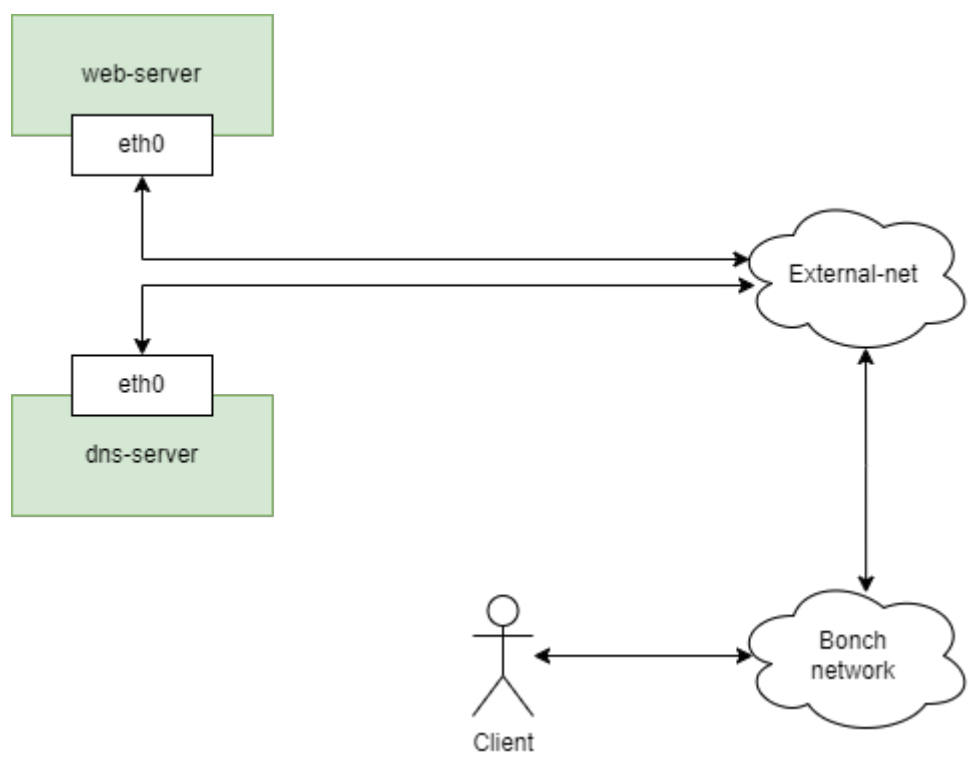


Рисунок 1. Схема стенда

- 1. Создать виртуальные машины для работы

Название виртуальной машины	Источник	Тип инстанса	Сети для внешнего подключения
web-server	Образ-Ubuntu-server20.04	small	external-net

Название виртуальной машины	Источник	Тип инстанса	Сети для внешнего подключения
dns-server	Образ-Ubuntu-server20.04	small	external-net

Так же нужно проверить развернутую инфраструктуру на соответствие схеме на рисунке 1.

“ В группах безопасности необходимо разрешить DNS

# 1. Установка и настройка авторитетного DNS сервера

Обновляем пакеты внутри системы до последней версии

```
sudo apt update
# Необязательно, но желательно
sudo apt full-upgrade -y
```

Устанавливаем сервер реализацию DNS сервера **PowerDNS**

```
sudo apt install pdns-server pdns-backend-sqlite3 sqlite3 -y
```

Настроить для работы с sqlite бд

```
sudo mkdir /var/lib/powerdns
sudo sqlite3 /var/lib/powerdns/pdns.sqlite3 < /usr/share/doc/pdns-backend-sqlite3/schema.sqlite3.sql
sudo chown -R pdns:pdns /var/lib/powerdns
```

Выключаем **systemd-resolved**

```
sudo systemctl disable --now systemd-resolved.service
```

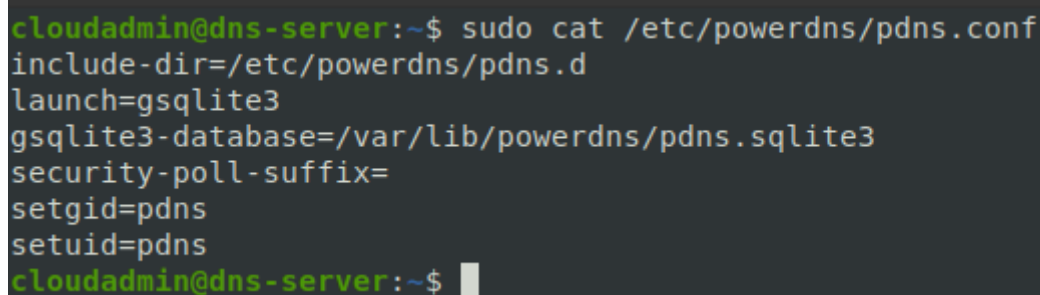
Изменяем используемый dns-сервер на **172.17.1.10**

```
sudo rm -rf /etc/resolv.conf
echo "nameserver 172.17.1.10" | sudo tee /etc/resolv.conf
```

Отредактировать **/etc/powerdns/pdns.conf**

```
include-dir=/etc/powerdns/pdns.d
launch=sqlite3
sqlite3-database=/var/lib/powerdns/pdns.sqlite3
security-poll-suffix=
setgid=pdns
setuid=pdns
```

1. **include-dir=/etc/powerdns/pdns.d**: Эта строка указывает серверу PowerDNS загружать все конфигурационные файлы из директории */etc/powerdns/pdns.d*. Это позволяет разделять конфигурацию на несколько файлов для более удобного управления.
2. **launch=sqlite3**: Эта строка определяет, какой бэкенд базы данных будет использоваться для хранения данных сервера DNS. В данном случае, указан бэкенд *sqlite3*, что означает использование базы данных SQLite.
3. **sqlite3-database=/var/lib/powerdns/pdns.sqlite3**: Здесь указывается путь к файлу базы данных SQLite, который будет использоваться сервером PowerDNS для хранения записей DNS.
4. **security-poll-suffix=**: Этот параметр связан с безопасностью и суффиксом для защиты от определенных видов атак. В данном случае, он не задан (пустое значение), в данном случае это означает не использовать варианты защиты.
5. **setgid=pdns** и **setuid=pdns**: Эти параметры устанавливают идентификатор группы и пользователя, от которого будет работать процесс PowerDNS. В этом случае, сервер будет запущен с правами группы *pdns* и пользователем *pdns* для обеспечения минимальных привилегий в целях безопасности.



```
cloudadmin@dns-server:~$ sudo cat /etc/powerdns/pdns.conf
include-dir=/etc/powerdns/pdns.d
launch=sqlite3
sqlite3-database=/var/lib/powerdns/pdns.sqlite3
security-poll-suffix=
setgid=pdns
setuid=pdns
cloudadmin@dns-server:~$
```

Рисунок 2. Пример конфига

Активируем демона dns-сервера

```
sudo systemctl enable --now pdns
```

Создаем зону по имени пользователя(для примера используется *devops-course.test*)

```
sudo pdnsutil create-zone {zone name} ns1.{zone name}
sudo pdnsutil add-record {zone name}. www A 127.0.0.1
```

```
cloudadmin@dns-server:~$ sudo -u pdns pdnsutil create-zone devops-course.test ns1.devops-course.test
Nov 13 12:28:45 [bindbackend] Done parsing domains, 0 rejected, 0 new, 0 removed
Creating empty zone 'devops-course.test'
Also adding one NS record
cloudadmin@dns-server:~$ sudo -u pdns pdnsutil add-record devops-course.test. www A 127.0.0.1
Nov 13 12:28:50 [bindbackend] Done parsing domains, 0 rejected, 0 new, 0 removed
New rrsset:
www.devops-course.test. 3600 IN A 127.0.0.1
```

Проверьте работу сервера

```
nslookup www.devops-course.test {server ip}
```

Запрос делается с пользовательского ПК

```
~$ nslookup www.devops-course.test 172.17.36.216
Server:      172.17.36.216
Address:     172.17.36.216#53

Name:   www.devops-course.test
Address: 127.0.0.1
```

Рисунок 3. Пример ответ при адресе сервера 172.17.36.216

## 2. Настроить DNS для работы в соответствии со стандартом RFC 2136

Для работы по стандарту rfc2136, необходимо изменить конфигурацию работы PowerDNS, в конфигурационном файле необходимо добавить строки:

```
dnsupdate=yes
allow-dnsupdate-from=172.16.0.0/12
```

1. **dnsupdate=yes** - это включает поддержку динамического обновления DNS-записей (DNSUPDATE), что позволяет клиентам обновлять записи на сервере DNS.
2. **allow-dnsupdate-from=172.16.0.0/12** - это определяет диапазон IP-адресов (в данном случае, подсеть) для разрешения выполнения динамических обновлений DNS. Только запросы, идущие от адресов в этой подсети (172.16.0.0 - 172.31.255.255), будут разрешены для выполнения обновлений.

```
cloudadmin@dns-server:~$ sudo cat /etc/powerdns/pdns.conf
include-dir=/etc/powerdns/pdns.d
launch=sqlite3
sqlite3-database=/var/lib/powerdns/pdns.sqlite3
security-poll-suffix=
setgid=pdns
setuid=pdns
#rfc2136
dnsupdate=yes
allow-dnsupdate-from=172.16.0.0/12
cloudadmin@dns-server:~$
```

Рисунок 4. Конфигурация после обновления

Создать TSIG ключ с именем **dnsupdater**

```
sudo pdnsutil generate-tsig-key dnsupdater hmac-sha512
```

pdnsutil - это утилита для работы с PowerDNS. generate-tsig-key - команда для создания нового ключа TSIG. dnsupdater - это имя, которое будет присвоено сгенерированному ключу TSIG. hmac-sha512 - это алгоритм хеширования, используемый для создания ключа.

```
cloudadmin@dns-server:~$ sudo pdnsutil generate-tsig-key dnsupdater hmac-sha512
Nov 13 13:40:04 [bindbackend] Done parsing domains, 0 rejected, 0 new, 0 removed
Create new TSIG key dnsupdater hmac-sha512 zAKnKxGWFnwRAzXlN1+qcKJGK9PiS6+gBw70RU5JLNl8xtHmvyQUtXeGHPWPg
B0mBSpNnMLVFPfgbosN2KsKLA==
cloudadmin@dns-server:~$
```

Рисунок 5. Пример корректного создания TSIG ключа

Посмотреть существующие ключи можно командой:

```
sudo pdnsutil list-tsig-keys
```

Для проверки работы rfc2136 добавим новую **A** запись **test** указывающую, на наш web-сервер

```
nsupdate <<!
server {ip dns-server} 53
zone {zone name}
update add test.{zone name} 3600 A {ip web-server}
key hmac-sha512:{key name} {tsig key}
send
!
```

выполнять с устройства в сети 172.16.0.0/12

```
nsupdate <<!  
server 172.17.36.216 53  
zone devops-course.test  
update add test.devops-course.test 3600 A 172.17.36.16  
key hmac-sha512:dnsupdater mSJiu5jNDzwa9WHqkpGhnW/33z1ksCwmPs0Ap5Y4b1ogFNsfPt5YzuoqXq4YtpLNADtLPs7kzb3W54RxLGefcA==  
send  
!
```

Проверить, что запись добавилась

```
nslookup test.{zone name} {ip dns-server}
```

```
nslookup test.devops-course.test 172.17.36.216  
Server:      172.17.36.216  
Address:     172.17.36.216#53  
  
Name:   test.devops-course.test  
Address: 172.17.36.16
```

### 3. Настройка рекурсивного DNS

Так как наша конфигурация подразумевает совместную работы рекурсивного и авторитетного сервера на одном виртуальном сервере, нам необходимо разграничить их по портам, так для авторитетного сервера, будет использовать порт **5353**

Для этого в конфигурации авторитетного сервера необходимо изменить слушаемые порт добавив в конфигурационный файл строки

```
local-port=5353
```

После изменения конфигурации необходимо перезапустить службу

```
sudo systemctl restart pdns
```

Проверьте, что теперь слушается только порт **5353**

```
ss -tulpan
```

Установите рекурсивный сервер DNS

```
sudo apt install pdns-recursor -y
```

Привести конфиг(*/etc/powerdns/recursor.conf*) рекурсивного DNS к виду:

```
config-dir=/etc/powerdns
hint-file=/usr/share/dns/root.hints
include-dir=/etc/powerdns/recursor.d
local-address=0.0.0.0
local-port=53
lua-config-file=/etc/powerdns/recursor.lua
public-suffix-list-file=/usr/share/publicsuffix/public_suffix_list.dat
quiet=yes
security-poll-suffix=
setgid=pdns
setuid=pdns

# authoritative zone
forward-zones={zone name}=127.0.0.1:5353
forward-zones-recurse={zone name}=127.0.0.1:5353

# also
forward-zones+=.=1.1.1.1
forward-zones-recurse+=.=1.1.1.1
```

1. **config-dir=/etc/powerdns** - указывает на папку с дополнительными файлами конфигурации для PowerDNS Recursor.
2. **hint-file=/usr/share/dns/root.hints** - определяет местоположение файла с базовыми подсказками (hints) для поиска корневых серверов DNS.
3. **include-dir=/etc/powerdns/recursor.d** - указывает на папку, где Recursor будет искать дополнительные файлы конфигурации для обработки.
4. **local-address=0.0.0.0** и **local-port=53** - определяют, на каком адресе и порту будет слушать Recursor для входящих DNS-запросов.
5. **lua-config-file=/etc/powerdns/recursor.lua** - указывает на файл конфигурации Lua, который может содержать дополнительные пользовательские настройки для Recursor.
6. **quiet=yes** - включает режим тишины, что означает меньше вывода информации в логах или на консоль.
7. **forward-zones={zone name}=127.0.0.1:5353** и **forward-zones-recurse={zone name}=127.0.0.1:5353** - эти строки настраивают перенаправление запросов для конкретной зоны {zone name} на адрес 127.0.0.1 с портом 5353.
8. **forward-zones+=.=1.1.1.1** и **forward-zones-recurse+=.=1.1.1.1** - эти строки добавляют общее перенаправление запросов для всех неопределенных явным образом зон на указанный IP-адрес (1.1.1.1).

```
cloudadmin@dns-server:~$ cat /etc/powerdns/recursor.conf
config-dir=/etc/powerdns
hint-file=/usr/share/dns/root.hints
include-dir=/etc/powerdns/recursor.d
local-address=0.0.0.0
local-port=53
lua-config-file=/etc/powerdns/recursor.lua
public-suffix-list-file=/usr/share/publicsuffix/public_suffix_list.dat
quiet=yes
security-poll-suffix=
setgid=pdns
setuid=pdns

# authoritative zone
forward-zones=devops-course.test=127.0.0.1:5353
forward-zones-recurse=devops-course.test=127.0.0.1:5353

# also
forward-zones+=.1.1.1.1
forward-zones-recurse+=.1.1.1.1

cloudadmin@dns-server:~$
```

Включаем и добавляем автозапуск демона

```
sudo systemctl enable pdns-recursor
sudo systemctl restart pdns-recursor
```

## 4. Выпуск сертификата безопасности

Переходим на узел веб сервера и используя команду генерируем самоподписанный сертификат для нашего будущего тестового сайта

```
openssl req -x509 -nodes -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -subj '/CN={Доменное имя}' -addext "subjectAltName = DNS:{Доменное имя}" -addext "keyUsage = digitalSignature, keyEncipherment" -addext "extendedKeyUsage = serverAuth"
```

```
cloudadmin@web-server:~$ openssl req -x509 -nodes -newkey rsa:4096 -keyout key.p
em -out cert.pem -days 365 -subj '/CN=test.devops-course.test' -addext "subjectA
ltName = DNS:test.devops-course.test" -addext "keyUsage = digitalSignature, keyE
ncipherment" -addext "extendedKeyUsage = serverAuth"
Generating a RSA private key
.....+++++
.....+++++
+
writing new private key to 'key.pem'
-----
```



## Пример создания сертификата

Для проверки созданного сертификата можно использовать команду

```
openssl x509 -in cert.pem -noout -text
```

# 5. Проверка сертификата безопасности

Для проверки мы установим nginx и установим на него ранее созданный сертификат. Нам необходимо начать с установки nginx

```
sudo apt update  
sudo apt install nginx -y
```

После установки необходимо конфигурацию по умолчанию (*/etc/nginx/sites-available/default*) и привести ее к виду:

```
server {  
    listen 80 default_server;  
    server_name {доменное имя};  
    return 301 https://$host$request_uri;  
}  
  
server {  
  
    listen 443 ssl http2;  
    server_name {доменное имя};  
  
    ssl_certificate      /home/cloudadmin/cert.pem;  
    ssl_certificate_key  /home/cloudadmin/key.pem;  
  
    ssl_session_cache    builtin:1000 shared:SSL:10m;  
    ssl_protocols TLSv1.3;  
    ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;  
    ssl_prefer_server_ciphers on;  
  
    access_log          /var/log/nginx/cloud.access.log;  
    error_log           /var/log/nginx/cloud.error.log;
```

```
root /var/www/html;

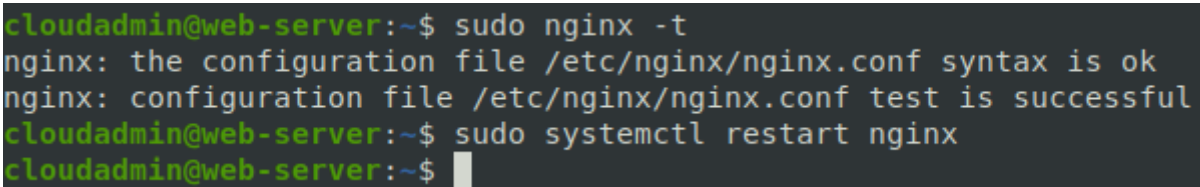
index index.html index.htm index.nginx-debian.html;

location / {
    try_files $uri $uri/ =404;
}

}
```

После этого можно проверить конфигурацию **nginx** и перезапустить демона

```
sudo nginx -t
sudo systemctl restart nginx
```



```
cloudadmin@web-server:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
cloudadmin@web-server:~$ sudo systemctl restart nginx
cloudadmin@web-server:~$
```

Пример ответа при правильной конфигурации

После этого необходимо убедиться в наличии созданного вами доменного имени с помощью утилиты **dig**

```
dig {доменное имя} @{адрес dns сервера}
```

```
cloudadmin@web-server:~$ dig test.devops-course.test @172.17.36.216

; <<>> DiG 9.16.1-Ubuntu <<>> test.devops-course.test @172.17.36.216
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30476
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:;, udp: 512
;; QUESTION SECTION:
;test.devops-course.test.      IN      A

;; ANSWER SECTION:
test.devops-course.test. 3600    IN      A      172.17.36.16

;; Query time: 4 msec
;; SERVER: 172.17.36.216#53(172.17.36.216)
;; WHEN: Чт ноя 16 13:46:54 MSK 2023
;; MSG SIZE rcvd: 68
```

Пример валидного ответа

После этого Вам необходимо изменить DNS сервер на вашем устройстве, на созданный вами DNS сервер и попробовать открыть в браузере ваш домен



Пример выполненной работы

---

Версия #17

Тарабанов Илья Федорович создал 13 ноября 2023 11:13:54

Тарабанов Илья Федорович обновил 16 мая 2024 19:12:47