

DevOps

- Настройка DNSMASQ под mint
- Права для папки .ssh
- Отключаем ввод пароля у Linux команд
- WireGuard
 - Работа с соединениями через консоль
 - Установка GUI для gnome
 - Установка сервера
 - dockvpn
- Настройка DNSMASQ под Fedora
- Как копировать Docker Images
- Альтернативы NGROK на PHP Expose
- S3FTP

Настройка DNSMASQ под mint

Установка DNS сервера

```
apt-get install dnsmasq resolvconf
```

Отключаем стандартный DNS-ресолвер systemd:

```
systemctl disable systemd-resolved  
systemctl stop systemd-resolved
```

Редактируем `/etc/NetworkManager/NetworkManager.conf` => добавляем в секцию "[main]" следующие строки:

```
dns=default  
rc-manager=resolvconf
```

Возможно нужно прописать в настройках соединения DNS **127.0.0.1** 8.8.8.8 8.8.4.4

`/etc/resolv.conf` должен содержать "nameserver 127.0.0.1", указывающий на локальный dnsmasq, а `/run/dnsmasq/resolv.conf` (часть настроек запущенного dnsmasq) — IP-адреса внешних ресолверов, которые NetworkManager получил по DHCP или из настроек сетевого соединения.

Если остановить dnsmasq вручную (systemctl stop dnsmasq), IP-адреса внешних ресолверов должны переместиться в `/etc/resolv.conf`

Конфиги

`/etc/dnsmasq.conf` dnsmasq.conf

`/etc/NetworkManager/dnsmasq.d/local.conf` local.conf

`/etc/NetworkManager/NetworkManager.conf` NetworkManager.conf

Полезные команды

`sudo lsof -i -P -n | grep LISTEN` Узнать кто занимает порты

`sudo /etc/init.d/dnsmasq restart` Перезагрузить dnsmasq

`service network-manager restart`

Полезные ссылки <https://cdnnow.ru/blog/dnslocal/>

Права для папки .ssh

Папка ./ssh - 700

id_rsa.pub - 644

config - 644

id_rsa - 600

Отключаем ввод пароля у Linux команд

```
sudo visudo
```

Вводим в самый конец строку в таком формате

```
seryak ALL = NOPASSWD: /usr/sbin/pm-suspend
```

Команду нужно вводить по прежнему через SUDO , но пароль больше не нужен

WireGuard

Работа с соединениями через консоль

```
sudo dnf install wireguard-tools
```

Редактировать соединение через GUI (в gnome 40+ нет поддержки WireGuard)

```
nm-connection-editor
```

Посмотреть все соединения

```
nmcli
```

Импортировать соединение из конфига

```
nmcli con import type wireguard file /home/seryak/peer2/peer2.conf
```

Выключить \ включить соединение

```
# nmcli connection up 'WireGuard connection 1'  
# nmcli connection down 'WireGuard connection 1'
```


Установка GUI для gnome

```
sudo apt install wireguard git dh-autoreconf libglib2.0-dev intltool build-essential libgtk-3-dev libnm-dev libsecret-1-dev network-manager-dev resolvconf
```

```
git clone https://github.com/max-moser/network-manager-wireguard
cd network-manager-wireguard
./autogen.sh --without-libnm-glib

./configure --without-libnm-glib --prefix=/usr --sysconfdir=/etc --libdir=/usr/lib/x86_64-linux-gnu --libexecdir=/usr/lib/NetworkManager --localstatedir=/var

make
sudo make install
```


Установка сервера

Docker-compose yaml

```
version: "2.1"
services:
  wireguard:
    image: ghcr.io/linuxserver/wireguard
    container_name: wireguard
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    environment:
      - PUID=1001
      - PGID=1001
      - TZ=Europe/Netherlands
      - SERVERURL=auto #optional
      - SERVERPORT=51820 #optional
      - PEERS=1 #optional
      - PEERDNS=auto #optional
      - INTERNAL_SUBNET=10.13.13.0 #optional
      - ALLOWEDIPS=0.0.0.0/0 #optional
    volumes:
      - /app/vpn/wireguard/config:/config
      - /lib/modules:/lib/modules
    ports:
      - 51820:51820/udp
    sysctls:
      - net.ipv4.conf.all.src_valid_mark=1
    restart: always
```

Рассмотрим основные параметры:

- **PUID=1000** - ID пользователя от которого запускаем контейнер. Если вы создавали пользователя с айдишником 1337 то вписываем сюда 1337
- **PGID=1000** - ID группы от которой запускаем контейнер. Если вы создавали пользователя с айдишником 1337 то вписываем сюда 1337
- **TZ=Europe/London** - Таймзона например Europe/Netherlands
- **SERVERURL=auto** - Если у нас есть домен можем вписать его сюда для удобства либо оставляем значение по умолчанию
- **SERVERPORT=51820** - Порт на котором будет висеть wireguard. Не забываем разрешить в фаерволле
- **PEERS=1** - Количество создаваемых конфигов. Хорошая практика 1 конфиг = 1 клиент, а не сажать 10 клиентов на 1 конфиг. При подключении каждый клиент будет иметь уникальный ip в виртуальной подсети
- **PEERDNS=auto** - Можно вписать кастомные днс или например 1.1.1.1 или оставить значение auto
- **INTERNAL_SUBNET=10.13.13.0** - Внутренняя виртуальная сеть
- **ALLOWEDIPS=0.0.0.0/0** - Диапазоны ip к которым пиры могут обращаться через VPN тунель. По стандарту все ip
- **/app/vpn/wireguard/config:/config** - Место для хранения данных докер контейнера

Управление сервером

`sudo docker exec -it wireguard wg` - статистика

`sudo docker-compose up -d --force-recreate` - для изменения конфига

`sudo docker exec -it wireguard /app/show-peer` НОМЕР-ПИРА - показать qr код

WireGuard

dockovpn

<https://dockovpn.io/ru/>

Настройка DNSMASQ под Fedora

/etc/default/dnsmasq:

```
IGNORE_RESOLVCONF=yes
```

Создаем файл `/etc/resolv.personal` с теми Dns которые нам нужны , например гугловские

```
nameserver 5.132.191.104  
nameserver 103.236.162.119
```

/etc/dnsmasq.conf

```
resolv-file=/etc/resolv.personal
```

Полезные команды

```
systemctl restart dnsmasq
```

```
systemctl restart NetworkManager
```


Как копировать Docker Images

```
docker save -o <path for generated tar file> <image name>
```

```
docker load -i <path to image tar file>
```


Альтернативы NGROK на PHP Expose

<https://expose.dev>

Установка

<https://expose.dev/docs/getting-started/installation>

Нужно скопировать бинарник (для запуска нужен PHP)

```
curl https://github.com/beyondcode/expose/raw/master/builds/expose -L --output expose
chmod +x expose
```

Серверный режим

```
./expose serve ngrok.wtolk.ru --port=3000
```

Указываем домен (будет работать на поддоменах) и порт. Как вариант можно использовать докер :

```
version: "3"
services:
  expose:
    image: beyondcodegmbh/expose-server:latest
    ports:
      - 3002:3002
    environment:
      port: 3002
      domain: tun.wtolk.ru
      username: seryak
      password: wtolk2210
```



```
restart: always
volumes:
- ./database/expose.db: /root/.expose
```

Админка доступна по домену `expose.domain` . Важно задать права на запись для базы SQLite

Пример конфига для вебсервера Caddy (указываем порт который включили в expose)

```
http://ngrok.wtolk.ru {
    reverse_proxy * http://php80:3000
}

http://*.ngrok.wtolk.ru {
    reverse_proxy * http://php80:3000
}
```

Клиентский режим

```
./expose share --subdomain=ola http://mesphp.web
```

Запускать нужно на основной машине (не в докере).

Пример конфиг файла (.expose.php рядом с бинарником)

```
<?php

return [

    /*
    | -----
    | Servers
    | -----
    |
    | The available Expose servers that your client can connect to.
    | When sharing sites or TCP ports, you can specify the server
    | that should be used using the `--server=` option.
```



```

|
*/
'servers' => [
  'main' => [
    'host' => 'tun.wtolk.ru',
    'port' => 3002,
  ],
],

/*
|-----
| Server Endpoint
|-----
|
| When you specify a server that does not exist in above static array,
| Expose will perform a GET request to this URL and tries to retrieve
| a JSON payload that looks like the configurations servers array.
|
| Expose then tries to load the configuration for the given server
| if available.
|
*/
'server_endpoint' => '',

/*
|-----
| Default Server
|-----
|
| The default server from the servers array,
| or the servers endpoint above.
|
*/
'default_server' => 'main',

/*
|-----
| DNS
|-----
|

```



```

| The DNS server to use when resolving the shared URLs.
| When Expose is running from within Docker containers, you should set this to
| `true` to fall-back to the system default DNS servers.
|
*/
'dns' => '127.0.0.1',

/*
| -----
| Auth Token
| -----
|
| The global authentication token to use for the expose server that you
| are connecting to. You can let expose automatically update this value
| for you by running
|
| > expose token YOUR-AUTH-TOKEN
|
*/
'auth_token' => '27b3a491-71e8-4a12-b6ee-87011e139de0',

/*
| -----
| Default Domain
| -----
|
| The custom domain to use when sharing sites with Expose.
| You can register your own custom domain using Expose Pro
| Learn more at: https://expose.dev/get-pro
|
| > expose default-domain YOUR-CUSTOM-WHITELABEL-DOMAIN
|
*/
'default_domain' => null,

/*
| -----
| Default TLD
| -----
|

```



```
| The default TLD to use when sharing your local sites. Expose will try
| to look up the TLD if you are using Laravel Valet automatically.
| Otherwise you can specify it here manually.
|
*/
'default_tld' => 'web',

/*
|-----
| Default HTTPS
|-----
|
| Whether to use HTTPS as a default when sharing your local sites. Expose
| will try to look up the protocol if you are using Laravel Valet
| automatically. Otherwise you can specify it here manually.
|
*/
'default_https' => false,

/*
|-----
| Maximum Logged Requests
|-----
|
| The maximum number of requests to keep in memory when inspecting your
| requests and responses in the local dashboard.
|
*/
'max_logged_requests' => 25,

/*
|-----
| Maximum Allowed Memory
|-----
|
| The maximum memory allocated to the expose process.
|
*/
'memory_limit' => '128M',
```



```

/*
| -----
| Skip Response Logging
| -----
|
| Sometimes, some responses don't need to be logged. Some are too big,
| some can't be read (like compiled assets). This configuration allows you
| to be as granular as you wish when logging the responses.
|
| If you run constantly out of memory, you probably need to set some of these up.
|
| Keep in mind, by default, BINARY requests/responses are not logged.
| You do not need to add video/mp4 for example to this list.
|
*/
'skip_body_log' => [
  /**
   * | Skip response logging by HTTP response code. Format: 4*, 5*.
   */
  'status' => [
    // "4*"
  ],
  /**
   * | Skip response logging by HTTP response content type. Ex: "text/css".
   */
  'content_type' => [
    //
  ],
  /**
   * | Skip response logging by file extension. Ex: ".js.map", ".min.js",
   * | ".min.css".
   */
  'extension' => [
    '.js.map',
    '.css.map',
  ],
  /**
   * | Skip response logging if response size is greater than configured value.
   * | Valid suffixes are: B, KB, MB, GB.
   * | Ex: 500B, 1KB, 2MB, 3GB.

```



```

    */
    'size' => '1MB',
],

'admin' => [

    /*
    |-----
    | Database
    |-----
    |
    | The SQLite database that your expose server should use. This database
    | will hold all users that are able to authenticate with your server,
    | if you enable authentication token validation.
    |
    */
    'database' => implode(DIRECTORY_SEPARATOR, [
        $_SERVER['HOME'] ?? __DIR__,
        '.expose',
        'expose.db',
    ]),

    /*
    |-----
    | Validate auth tokens
    |-----
    |
    | By default, once you start an expose server, anyone is able to connect to
    | it, given that they know the server host. If you want to only allow the
    | connection from users that have valid authentication tokens, set this
    | setting to true. You can also modify this at runtime in the server
    | admin interface.
    |
    */
    'validate_auth_tokens' => false,

    /*
    |-----
    | TCP Port Sharing
    |-----

```



```

|
| Control if you want to allow users to share TCP ports with your Expose
| server. You can add fine-grained control per authentication token,
| but if you want to disable TCP port sharing in general, set this
| value to false.
|
|
*/
'allow_tcp_port_sharing' => true,

/*
|-----
| TCP Port Range
|-----
|
| Expose allows you to also share TCP ports, for example when sharing your
| local SSH server with the public. This setting allows you to define the
| port range that Expose will use to assign new ports to the users.
|
| Note: Do not use port ranges below 1024, as it might require root
| privileges to assign these ports.
|
|
*/
'tcp_port_range' => [
  'from' => 50000,
  'to' => 60000,
],

/*
|-----
| Maximum connection length
|-----
|
| If you want to limit the amount of time that a single connection can
| stay connected to the expose server, you can specify the maximum
| connection length in minutes here. A maximum length of 0 means that
| clients can stay connected as long as they want.
|
|
*/
'maximum_connection_length' => 0,

```



```

/*
|-----
| Maximum number of open connections
|-----
|
| You can limit the amount of connections that one client/user can have
| open. A maximum connection count of 0 means that clients can open
| as many connections as they want.
|
| When creating users with the API/admin interface, you can
| override this setting per user.
|
*/
'maximum_open_connections_per_user' => 0,

/*
|-----
| Subdomain
|-----
|
| This is the subdomain that your expose admin dashboard will be available at.
| The given subdomain will be reserved, so no other tunnel connection can
| request this subdomain for their own connection.
|
*/
'subdomain' => 'expose',

/*
|-----
| Reserved Subdomain
|-----
|
| Specify any subdomains that you don't want to be able to register
| on your expose server.
|
*/
'reserved_subdomains' => [],

/*
|-----

```



```

| Subdomain Generator
| -----
|
| This is the subdomain generator that will be used, when no specific
| subdomain was provided. The default implementation simply generates
| a random string for you. Feel free to change this.
|
| */
| 'subdomain_generator' =>
\App\Server\SubdomainGenerator\RandomSubdomainGenerator::class,

/*
| -----
| Users
| -----
|
| The admin dashboard of expose is protected via HTTP basic authentication
| Here you may add the user/password combinations that you want to
| accept as valid logins for the dashboard.
|
| */
| 'users' => [
|     'username' => 'password',
| ],

/*
| -----
| User Repository
| -----
|
| This is the user repository, which by default loads and saves all authorized
| users in a SQLite database. You can implement your own user repository
| if you want to store your users in a different store (Redis, MySQL, etc.)
|
| */
| 'user_repository' => \App\Server\UserRepository\DatabaseUserRepository::class,

| 'subdomain_repository' =>
\App\Server\SubdomainRepository\DatabaseSubdomainRepository::class,

```



```

'logger_repository' => \App\Server\LoggerRepository\NullLogger::class,

/*
|-----
| Messages
|-----
|
| The default messages that the expose server will send the clients.
| These settings can also be changed at runtime in the expose admin
| interface.
|
*/
'messages' => [
    'resolve_connection_message' => function ($connectionInfo, $user)
{
    return config('expose.admin.messages.message_of_the_day');
},

    'message_of_the_day' => 'Thank you for using expose.',

    'invalid_auth_token' => 'Authentication failed. Please check your authentication
token and try again.',

    'subdomain_taken' => 'The chosen subdomain :subdomain is already taken. Please
choose a different subdomain.',

    'subdomain_reserved' => 'The chosen subdomain :subdomain is not available. Please
choose a different subdomain.',

    'custom_subdomain_unauthorized' => 'You are not allowed to specify custom
subdomains. Please upgrade to Expose Pro. Assigning a random subdomain instead.',

    'custom_domain_unauthorized' => 'You are not allowed to use this custom
domain.',

    'tcp_port_sharing_unauthorized' => 'You are not allowed to share TCP ports.
Please upgrade to Expose Pro.',

    'no_free_tcp_port_available' => 'There are no free TCP ports available on this
server. Please try again later.',

```



```
        'tcp_port_sharing_disabled' => 'TCP port sharing is not available on this Expose
server.',
    ],

    'statistics' => [
        'enable_statistics' => true,

        'interval_in_seconds' => 3600,

        'repository' =>
\App\Server\StatisticsRepository\DatabaseStatisticsRepository::class,
    ],
],
];
```


S3FTP

<https://cloud.yandex.ru/docs/storage/tools/sftp>

Команда для генерации сертификатов

```
openssl req -x509 -nodes -days 9000 -newkey rsa:2048 -keyout ftp.key -out ftp.pem
```

Структура

secrets/credentials - создается автоматически

secrets/ftp.key

secrets/ftp.pem

env.list

docker-compose.yml

Пример env.list

```
S3_BUCKET=sites-storages: arka1m19.ru - имя бакета и папка через двоеточие
SFTP=NO
FTP=YES
FTP_USER=arka - Логин
FTP_PASS=fer - Пароль
FTP_PASV_ADDRESS=localhost
FTP_PASV_ENABLE=YES
FTP_PASV_PROMISCUOUS=YES
FTP_PORT_PROMISCUOUS=YES
FTP_SSL_ENABLE=YES
```

Пример docker-compose.yml

```
version: '2'
# Список сервисов (контейнеров)
services:
```


ftp:

image: cr.yandex/crp9ftr22d26age3hulg/ftp-s3-gateway: 1.0

restart: unless-stopped

cap_add:

- SYS_ADMIN

security_opt:

- apparmor: unconfined

devices:

- /dev/fuse: /dev/fuse

env_file: env.list

ports:

- "1021: 21"
- "21100: 21100"

volumes:

- ./secrets: /secrets