NGINX

Для работы серверных компонентов системы на ОС Linux требуется веб-сервер, работающий в режиме работы обратного прокси-сервера.
Nginx — это HTTP-сервер и обратный прокси-сервер, почтовый прокси-сервер, а также TCP/UDP прокси-сервер общего назначения, продукт позиционируется производителем как простой, быстрый и надежный.

> Для установки NGINX должен быть подключен и настроен репозиторий пакетов nginx. Если это не было сделано автоматически, добавьте репозиторий вручную.

Ручное добавление репозитория nginx

RHEL и производные дистрибутивы

Установите пакеты, необходимые для подключения yum-репозитория:

```
sudo yum install yum-utils
```

Для подключения yum-репозитория создайте файл с именем */etc/yum.repos.d/nginx.repo* со следующим содержимым:

```
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true

[nginx-mainline]
name=nginx mainline repo
baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
gpgcheck=1
enabled=0
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true
```

Debian и производные дистрибутивы

Установите пакеты, необходимые для подключения apt-репозитория:

Ubuntu:

```
sudo apt install curl gnupg2 ca-certificates lsb-release ubuntu-keyring
```

Debian:

```
sudo apt install curl gnupg2 ca-certificates lsb-release debian-archive-keyring
```

Импортируйте официальный ключ, используемый apt для проверки подлинности пакетов:

```
curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor | sudo tee /usr/share/keyrings
/nginx-archive-keyring.gpg >/dev/null
```

Для подключения apt-репозитория выполните следующую команду:

```
echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] http://nginx.org/packages
/ubuntu `lsb_release -cs` nginx" | sudo tee /etc/apt/sources.list.d/nginx.list
```

Установка nginx

Чтобы установить nginx, выполните следующую команду:

RHEL и производные дистрибутивы:

```
sudo yum install nginx
```

В случае запроса подтверждения GPG-ключа проверьте, что отпечаток ключа совпадает с 573B FD6B 3D8F BC64 1079 A6AB ABF5 BD82 7BD9 BF62.

Debian и производные дистрибутивы:

```
sudo apt update
sudo apt install nginx
```

Документация по установке на прочие ОС доступна на официальном портале продукта.

Выпуск SSL/TLS сертификата

Для настройки защищенного соединения необходимо выпустить SSL/TLS сертификат на имя машины с установленным nginx. Возможно использовать самоподписанный сертификат или сертификат с УЦ.

Самоподписанный сертификат

1. Создайте самоподписанный сертификат утилитой openssl (вместо *SERVERNAME.DOMAIN* подставьте DNS-имя рабочей станции с nginx):

```
sudo openssl req -x509 -nodes -addext "subjectAltName=DNS:SERVERNAME.DOMAIN,DNS:www.SERVERNAME.DOMAIN" -days 730 -newkey rsa:2048 -keyout /etc/ssl/private/SSL.key -out /etc/ssl/private/SSL.crt
```

2. Добавьте сертификат в список доверенных на локальной машине в соответствии с выбранной для настройки ОС.

Для RHEL и производных дистрибутивов:

```
sudo cp /etc/ssl/private/SSL.crt /etc/pki/ca-trust/source/anchors/SSL.crt
sudo update-ca-trust extract
```

Для Debian и производных дистрибутивов:

```
sudo cp /etc/ssl/private/SSL.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates -f
```

3. Сделайте сертификат доверенным в домене, например, с помощью групповых политик.

Выпуск сертификата на УЦ

1. Выпустите сертификат на УЦ, например на Microsoft CA, экспортируйте данный сертификат в формате *.pfx* (с закрытым ключом, с цепочкой корневых /промежуточных УЦ) на рабочую станцию с установленным nginx.

⚠ **Субъект** (Subject) сертификата должен содержать FQDN сервера Indeed CM. **Дополнительное имя субъекта** (Subject Alternative Name) сертификата должно содержать атрибут **DNS-имя** (DNS Name) (FQDN сервера Indeed CM). Например: *redos.demo.local* или соответствующую запись с подстановочными знаками, например: **. demo.local* (Wildcard certificate). **Улучшенный ключ** (Enhanced Key Usage) сертификата должен содержать значение **Проверка подлинности сервера** (Server Authentication).

2. Добавьте сертификат корневого УЦ в доверенные на машине с установленным nginx.

Для RHEL и производных дистрибутивов:

```
sudo cp ./root-ca.crt /etc/pki/ca-trust/source/anchors/
sudo update-ca-trust extract
```

Для Debian и производных дистрибутивов:

```
sudo cp ./root-ca.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates -f
```

3. Разделите *.pfx* сертификат на файл цепочки сертификатов и ключ, сделайте файл ключа без пароля (необходимо подставить имя импортированного файла вместо *PFXFILE*):

```
openssl pkcs12 -in PFXFILE.pfx -chain -nokeys | sed -ne '/-BEGIN CERTIFICATE/,/END CERTIFICATE/p' > SSL.crt
openssl pkcs12 -in PFXFILE.pfx -nocerts -out SSLencrypted.key
openssl rsa -in SSLencrypted.key -out SSL.key
```

Файл цепочки сертификатов должен быть следующего вида:

```
-----BEGIN CERTIFICATE-----
#Ваш сертификат#
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
#Промежуточный сертификат#
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
#Корневой сертификат#
-----END CERTIFICATE-----
```

4. Скопируйте файлы цепочки сертификатов и ключа в папку, которая указана в файле конфигурации nginx:

```
sudo cp ./SSL.crt /etc/ssl/private/
sudo cp ./SSL.key /etc/ssl/private/
```

Настройка конфигурационного файла

Для работы Indeed CM требуется настроить nginx, чтобы он обслуживал запросы и отправлял их на проксируемый адрес (сервис Indeed CM).
Работа nginx и его модулей определяется в конфигурационном файле, по умолчанию он называется ***nginx.conf*** и в зависимости от операционной системы расположен в каталоге */usr/local/nginx/conf*, */etc/nginx* или */usr/local/etc/nginx*.

Таблица рекомендуемых к использованию директив:

Контекст	Директива	Значение по умолчанию	Рекомендуемое значение	Комментарий
http	*proxy_buffer_size*	4k|8k	16k	Увеличивается размер прокси буферов для передачи необходимой информации в http-запросах.
	proxy_buffers	8 4k | 8 8k	4 16k	Увеличивается размер прокси буферов для передачи необходимой информации в http-запросах.
	types_hash_max_size	1024	4096	Увеличивается размер хэш-таблиц для хранения информации в виду большого количества проксируемых сервисов.
	client_max_body_size	1m	10m	Увеличивается допустимый размер загружаемых в систему файлов.
server	*listen*	80	443 ssl	Изменяется порт прослушивания на протокол HTTPS, по умолчанию nginx настроен на HTTP.
			3003 ssl	Порт 3003 указывается для дополнительного контекста server в случае использования агентского функционала Indeed CM.
	server_name	—	*	* Требуется указать, обычно совпадает с именем машины, на которой установлен nginx. Используется для фильтра обработки запросов.
	ssl_certificate	—	*/etc/ssl/private/SSL.crt*	Для работы по HTTPS указывается путь к файлу с цепочкой сертификатов (SSL сертификат, сертификаты промежуточного и корневого УЦ).
	ssl_certificate_key	—	*/etc/ssl/private/SSL.key*	Для работы по HTTPS указывается путь к закрытому ключу SSL сертификата.
	ssl_verify_client	off	optional_no_ca	Добавляется в случае авторизации по сертификату (используется клиентскими агентами)

location	*proxy_pass*	—	*	* Один экземпляр контекста *location* направляет запросы на один адрес — сервис Indeed CM. Таким образом, контекстов *location* должно быть столько, сколько есть сервисов Indeed CM. Точка проксирования указывается в формате: http://localhost:*PORT*/cm/*SERVICENAME* http://localhost:*PORT*/*AGENTSERVICENAME* Где необходимо указать PORT — порт, на котором запущен сервис Indeed CM, а также SERVICENAME и AGENTSERVICENAME — имя запущенного сервиса.
	include	—	*/etc/nginx/conf.d/proxy.conf*	Некоторые директивы описываются для каждого *location*, и для компактности конфигурационного файла рекомендуется создать файл с часто используемым набором директив и подключать его в каждый *location* вместо описывания набора целиком.
	proxy_http_version	1.0	1.1	Версия 1.1 рекомендуется для keepalive подключений и NTLM аутентификации.
	proxy_cache_bypass	—	$http_upgrade	Определяет условия, при которых ответ не будет браться из кэша.
	proxy_set_header	—	Upgrade $http_upgrade	Определяет переход с HTTP/1.1 на Web-socket после установления соединения.
			Connection keep-alive	Для использования keepalive подключений.
			Host $host	Для сохранения в заголовках имени nginx сервера при их передаче сервисам Indeed CM.
			X-Real-IP $remote_addr	По умолчанию работа в режиме обратного прокси использует нестандартные заголовки о пользовательском IP адресе, что требует задания данной директивы.
			X-Forwarded-For $proxy_add_x_forwarded_for	Подобно *X-Real-IP $remote_addr*, определяет формирование заголовка для корректного проксирования. Если поле X-Forwarded-For не присутствовало в изначальном заголовке, то $proxy_add_x_forwarded_for = $remote_addr.

			X-Forwarded-Proto $scheme	Веб-сервер принимает запросы по HTTPS и проксирует их к HTTP сервисам Indeed CM для корректной подмены протокола.
	fastcgi_buffers	8 4k|8k	16 16k	Определяет количество и размер буферов для чтения ответов от FastCGI сервера, на одно подключение.
	fastcgi_buffer_size	4k|8k	32k	Определяет размер буфера для чтения первой части ответа от FastCGI сервера.
	proxy_set_header	—	x-ssl-client-cert $ssl_client_escaped_cert	Директива передавать клиентский сертификат при проксировании. Используется клиентскими агентами для авторизации по сертификату.

Вследствие использования в конфигурации многократного описания контекстов *location*, определенный набор директив будет повторяться. Для удобства конфигурации рекомендуется вынести данный набор в отдельный файл, а в описании контекста включать директивы из данного файла (директива *include*).

1. Создайте файл с многократно используемыми директивами. Возможно разместить такой файл с расширением *.conf* в каталоге */etc/nginx/conf.d/*.
   Рекомендуемое содержимое файла:

   ```
   proxy_http_version 1.1;
   proxy_set_header Upgrade $http_upgrade;
   proxy_set_header Connection keep-alive;
   proxy_set_header Host $host;
   proxy_cache_bypass $http_upgrade;
   proxy_set_header X-Real-IP $remote_addr;
   proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
   proxy_set_header X-Forwarded-Proto $scheme;
   fastcgi_buffers 16 16k;
   fastcgi_buffer_size 32k;
   ```

2. Сконфигурируйте основной файл конфигурации nginx.

   > Имена контекстов *location* должны совпадать с путем к проксируемому сервису.

   Пример файла nginx.conf, сконфигурированного для работы с Indeed CM:

   ```
   user nginx;
   worker_processes auto;
   error_log /var/log/nginx/error.log notice;
   events {    worker_connections 1024;    }

   http {
     proxy_buffer_size   16k;
     proxy_buffers   4 16k;
   ```

```
  types_hash_max_size  4096;
  client_max_body_size 10m;
  add_header X-Frame-Options SAMEORIGIN always;
  add_header X-Content-Type-Options nosniff;

    log_format  main  '[$time_local] $remote_addr VIA $scheme --- $status --- $request \n
$ssl_client_fingerprint';
  access_log  /var/log/nginx/access.log  main;

  sendfile            on;
  tcp_nopush          on;
  include             /etc/nginx/mime.types;
  default_type        application/octet-stream;

  server {
    listen          443 ssl;
    server_name       redos.demo.local;

    ssl_certificate    "/etc/ssl/private/SSL.crt";
    ssl_certificate_key "/etc/ssl/private/SSL.key";

         location /cm/mc
        {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5001/cm/mc;   }
    location /cm/ss
        {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5002/cm/ss;   }
    location /cm/rss
        {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5003/cm/rss;   }
    location /cm/api
            {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5004/cm/api;   }
    location /cm/credprovapi
        {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5005/cm/credprovapi;   }
        location /cm/oidc
        {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5008/cm/oidc;   }
      location /cm/wizard
        {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5009/cm/wizard;  }
     }
  server {
    listen          3003 ssl;
    server_name       redos.demo.local;

    ssl_certificate    "/etc/ssl/private/SSL.crt";
    ssl_certificate_key "/etc/ssl/private/SSL.key";
    ssl_verify_client   optional_no_ca;

    location /agentregistrationapi
            {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5006/agentregistrationapi;  }
```

```
    location /agentserviceapi
        {  include /etc/nginx/conf.d/proxy.conf;
                proxy_pass http://localhost:5007/agentserviceapi;
                proxy_set_header x-ssl-client-cert $ssl_client_escaped_cert;  }
  }
}
```

Изменения, сделанные в конфигурационном файле, не будут применены, пока nginx не будет отправлена команда перезагрузить конфигурацию или он не будет перезапущен. Для перезагрузки конфигурации выполните команду:

```
nginx -s reload
```

File	Modified
File proxy.conf	Jun 07, 2023 by Vladimir Gololobov
File nginx.conf	Sep 20, 2023 by Vladimir Gololobov

Download All