

Настройка аутентификации пользователей

По умолчанию в Jitsi Meet любой пользователь может создавать конференции. Если сервер развернут внутри корпоративной сети, тогда такой вариант может вам подойти. Но если разворачивать публичный сервер, необходима авторизация. Мы настроим Jitsi Meet так, что создавать конференции смогут только зарегистрированные пользователи, а подключаться к ним — кто угодно.

1. Сначала посмотрим основные каталоги, куда «прописался» при установке Jitsi вместе с компонентами:

`/etc/apt/sources.list.d` (там есть файл **jitsi-stable.list** — это файл с записью о добавленном репозитории Jitsi)

`/etc/apt` (там в файле **sources.list** одна из записей указывает на репозиторий Prosody)

`/etc/jitsi`

`/etc/prosody`

`/etc/nginx`

`/var/lib/prosody`

`/usr/lib/prosody`

`/usr/share/lua`

`/usr/share/lua5.1`

`/usr/share/lua5.2`

`/usr/share/keyrings` (там находится файл **jitsi-keyring.gpg** с ключом, заданным ещё до установки Jitsi)

`/usr/share/jicofo`

`/usr/share/jitsi-meet`

`/usr/share/jitsi-meet-prosody`

`/usr/share/jitsi-meet-turnserver`

`/usr/share/jitsi-meet-web-config`

`/usr/share/jitsi-videobridge`

2. Для того, чтобы программа могла правильно работать с пользователями, которые будут создавать конференции — нужен специальный компонент **luaunbound**.

В **Ubuntu 20.04** данный компонент нужно установить с помощью специального менеджера пакетов для языка программирования Lua. Этот менеджер называется **LuaRocks**.

Он должен был установиться по умолчанию вместе с самим языком Lua. Если этого не произошло, можно установить его вручную командами: `sudo apt install luarocks`

Далее выполняем ещё такие команды:

```
sudo apt install liblua5.1-0-dev liblua5.2-dev liblua50-dev
```

```
sudo apt install luaunbound-dev
```

После этого будет установлен инструмент командной строки с одноимённым названием.

Теперь с его помощью устанавливаем компонент **luaunbound**:

```
sudo luarocks install luaunbound
```

Теперь переходим непосредственно к настройкам аутентификации пользователей.

3. Отредактируем файл:

```
sudo nano /etc/prosody/conf.avail/media3.nbics.net.cfg.lua
```

В разделе **VirtualHost «[имя_домена]»** заменим анонимную аутентификацию на аутентификацию с паролем:

```
VirtualHost "media3.nbics.net"
```

```
authentication = "internal_hashed"
```

Было:

```
}  
  
VirtualHost "media3.nbics.net"  
  authentication = "jitsi-anonymous" -- do not delete me  
  -- Properties below are modified by jitsi-meet-tokens package config  
  -- and authentication above is switched to "token"  
  app_id="example_app_id"
```

Стало:

```
VirtualHost "media3.nbics.net"  
  authentication = "internal_hashed" -- do not delete me  
  -- Properties below are modified by jitsi-meet-tokens pa  
  -- and authentication above is switched to "token"  
  app_id="example_app_id"
```

Сохраняем файл клавишами:

Ctrl+X

Y

Enter

4. Теперь включим анонимный доступ для гостей, чтобы они могли подключаться к уже созданным видеоконференциям. Для этого в самый конец файла (после всего текста) добавим следующий блок:

```
VirtualHost "guest.media3.nbics.net"
```

```
authentication = "anonymous"
```

```
c2s_require_encryption = false
```

```
VirtualHost "guest.media3.nbics.net"  
authentication = "anonymous"  
c2s_require_encryption = false
```

В данном случае поддомен guest необходим только для настроек Jitsi. Для него не нужно создавать DNS-запись, генерировать TLS-сертификат или настраивать веб-сервер.

Сохраняем файл.

5. Далее будем редактировать файл:

```
sudo nano /etc/jitsi/meet/media3.nbics.net-config.js
```

Нужно найти в нем строки:

```
hosts: {
```

```
    // XMPP domain.
```

```
    domain: 'media3.nbics.net',
```

И добавить строку anonymousdomain: 'guest.media3.nbics.net ', чтобы в итоге получилось так:

```
hosts: {
```

```
    // XMPP domain.
```

```
    domain: 'media3.nbics.net',
```

```
    anonymousdomain: 'guest.media3.nbics.net ',
```

Тем самым мы указываем, на каком хосте нужно выполнять аутентификацию анонимных пользователей.

Было:

```
hosts: {
  // XMPP domain.
  domain: 'media3.nbics.net',

  // When using authentication, domain for guest users.
  // anonymousdomain: 'guest.example.com',
}
```

Стало:

```
hosts: {
  // XMPP domain.
  domain: 'media3.nbics.net',
  anonymousdomain: 'guest.media3.nbics.net',

  // When using authentication, domain for guest users.
  // anonymousdomain: 'guest.example.com',
}
```

Сохраняем файл.

6. Следующим шагом редактируем файл `/etc/jitsi/jicofo/jicofo.conf`

Изначально доступ в каталог **jicofo** ограничен, поэтому даём права на этот каталог командой:

```
sudo chmod a+x /etc/jitsi/jicofo/
```

Открываем файл:

```
sudo nano /etc/jitsi/jicofo/jicofo.conf
```

Найдём в нем раздел **jicofo** и добавим в него новый блок:

```
authentication: {
  enabled: true
  type: XMPP
  login-url: media3.nbics.net
}
```

Было:

```
GNU nano 4.8 jicofo.conf
# Jicofo HOCON configuration. See reference.conf in /usr/share/jicofo/jicofo.jar for
#available options, syntax, and default values.
jicofo {
  xmpp: {
    client: {
      client-proxy: "focus.media3.nbics.net"
      xmpp-domain: "media3.nbics.net"
      domain: "auth.media3.nbics.net"
      username: "focus"
      password: "jmJvAsDAGFAoBrBv"
    }
    trusted-domains: [ "recorder.media3.nbics.net" ]
  }
  bridge: {
    brewery-jid: "JvbBrewery@internal.auth.media3.nbics.net"
  }
}
```

Стало:

```
GNU nano 4.8 jicofo.conf
# Jicofo HOCON configuration. See reference.conf in /usr/share/jicofo/jicofo.jar for
#available options, syntax, and default values.
jicofo {
  xmpp: {
    client: {
      client-proxy: "focus.media3.nbics.net"
      xmpp-domain: "media3.nbics.net"
      domain: "auth.media3.nbics.net"
      username: "focus"
      password: "jmJvAsDAGFAoBrBv"
    }
    trusted-domains: [ "recorder.media3.nbics.net" ]
  }
  bridge: {
    brewery-jid: "JvbBrewery@internal.auth.media3.nbics.net"
  }
  authentication: {
    enabled: true
    type: XMPP
    login-url: media3.nbics.net
  }
}
```

7. Настройка Jitsi Meet завершена. Нам остается только создать пользователей, которые смогут создавать конференции. Для этого выполним команду (самостоятельно придумываем и вписываем в команду логин и пароль, без угловых скобок, домен тоже здесь лишь для примера):

```
sudo prosodyctl register <логин> media2.nbics.net <пароль>
```

8. Осталось перезапустить все компоненты Jitsi, чтобы применить выполненные настройки:

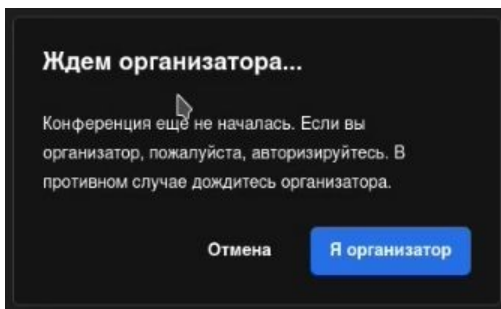
```
sudo systemctl restart prosody jicofo jitsi-videobridge2
```

Проверим работу всех служб:

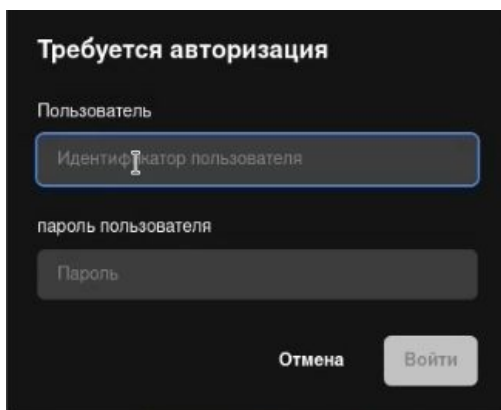
```
sudo systemctl status nginx prosody jicofo jitsi-videobridge2
```

Если работают, значит всё нормально.

9. Запускаем сайт. Если при создании конференции или переходе по ссылке-приглашению (в то время пока организатор ещё не подключился) появляется такое окно, значит всё настроено правильно:



10. Жмём кнопку «Я организатор», вписываем ранее заданный логин и пароль:



Если всё верно — попадаем в конференцию.