

# Установка Jitsi

## 1. Открытие портов брандмауэра (откроем их ещё и через iptables)

Для работы Jitsi Meet на сервере должны быть открыты несколько обязательных портов:

**80/TCP** – Если используется Let’s Encrypt, этот порт нужен для выпуска и обновления TLS-сертификата.

**443/TCP** – Для общего доступа к Jitsi Meet, в том числе через браузер.

**10000/UDP** – Для проведения аудио/видео конференций.

**5349/TCP** – Резервный порт для аудио/видео конференций на случай, если UPD-порт 10000 заблокирован.

Чтобы открыть эти порты, выполним команды:

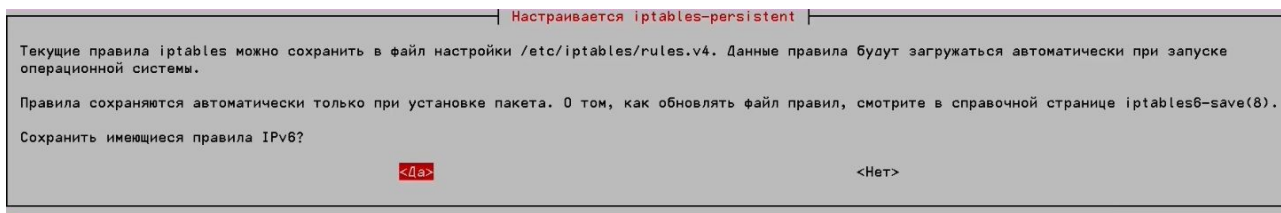
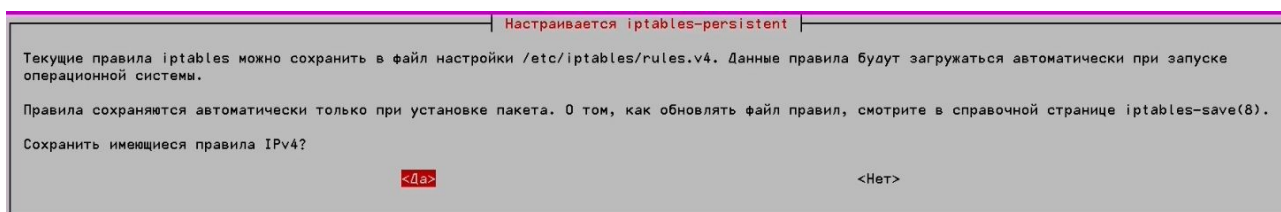
```
sudo iptables -I INPUT -p tcp --match multiport --dports 80,443 -j ACCEPT
```

```
sudo iptables -I INPUT -p tcp --dport 10000 -j ACCEPT
```

```
sudo iptables -I INPUT -p tcp --dport 5349 -j ACCEPT
```

## 2. Чтобы эти настройки сохранились и после перезагрузки сервера и брандмауэр не блокировал подключения, установим пакет iptables-persistent и сохраним правила:

```
sudo apt install iptables-persistent
```



```
sudo netfilter-persistent save
```

```
ubs@ubspc:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
ubs@ubspc:~$
```

3. Далее нужно настроить сервер, чтобы также установить соответствие доменного имени и IP-адреса. Выполним команду:

```
sudo hostnamectl set-hostname media3.nbics.net
```

Не забываем, что доменное имя нужно вписывать своё — это имя здесь для примера.

4. Добавляем репозиторий для Prosody. Prosody - это кроссплатформенный XMPP-сервер, который используется в Jitsi «под капотом»:

```
echo deb http://packages.prosody.im/debian $(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list
```

5. Ключ для Prosody:

```
wget https://prosody.im/files/prosody-debian-packages.key -O- | sudo apt-key add -
```

6. Устанавливаем интерпретатор скриптового языка Lua (на этом языке написан Prosody)

```
sudo apt install lua5.2
```

7. Добавляем ключ для Jitsi (это одна команда, хоть и длинная):

```
curl https://download.jitsi.org/jitsi-key.gpg.key | sudo sh -c 'gpg --dearmor > /usr/share/keyrings/jitsi-keyring.gpg'
```

8. Добавляем репозиторий Jitsi (тоже одна длинная команда):

```
echo 'deb [signed-by=/usr/share/keyrings/jitsi-keyring.gpg] https://download.jitsi.org stable/' | sudo tee /etc/apt/sources.list.d/jitsi-stable.list > /dev/null
```

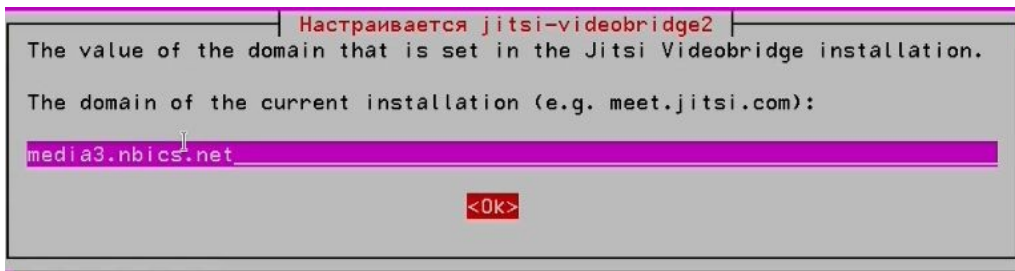
9. Обновляем список пакетов:

```
sudo apt update
```

10. Устанавливаем Jitsi Meet:

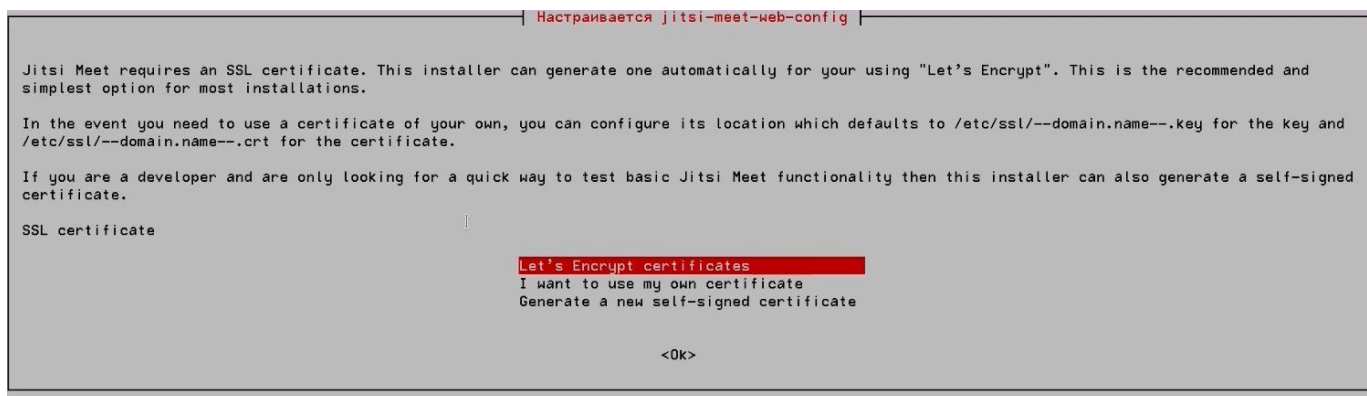
```
sudo apt install jitsi-meet
```

11. Во время установки в первом настроечном окне введём имя нашего домена:

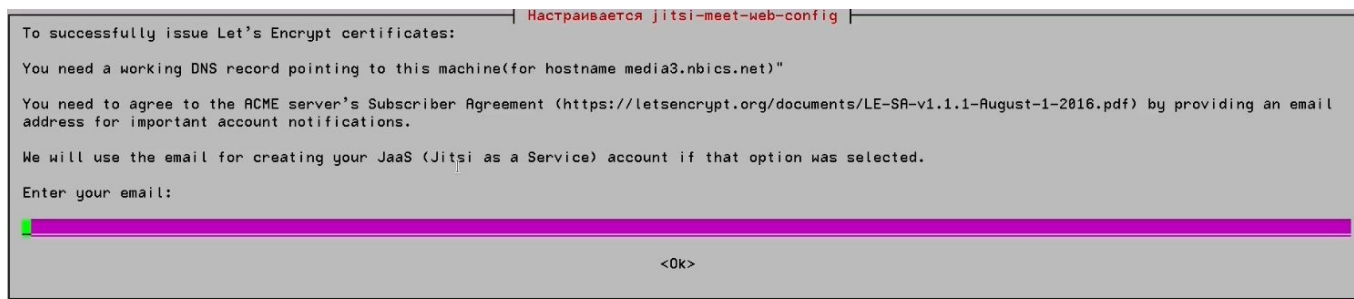


## 12. Выбираем первый пункт для настройки сертификата через Центр сертификации Lets

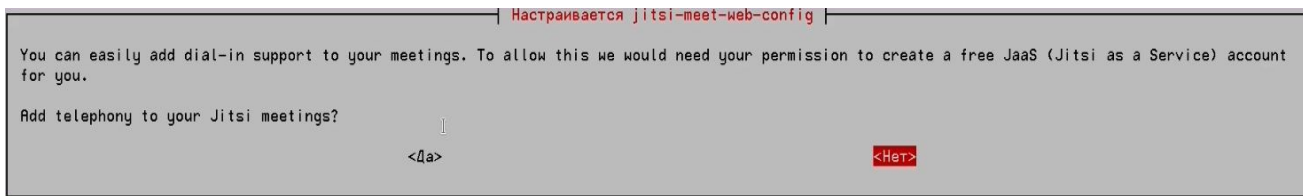
Encrypt:



## 13. Почтовый адрес (e-mail) вводить не обязательно:



## 14. Тут смело выбираем Нет:

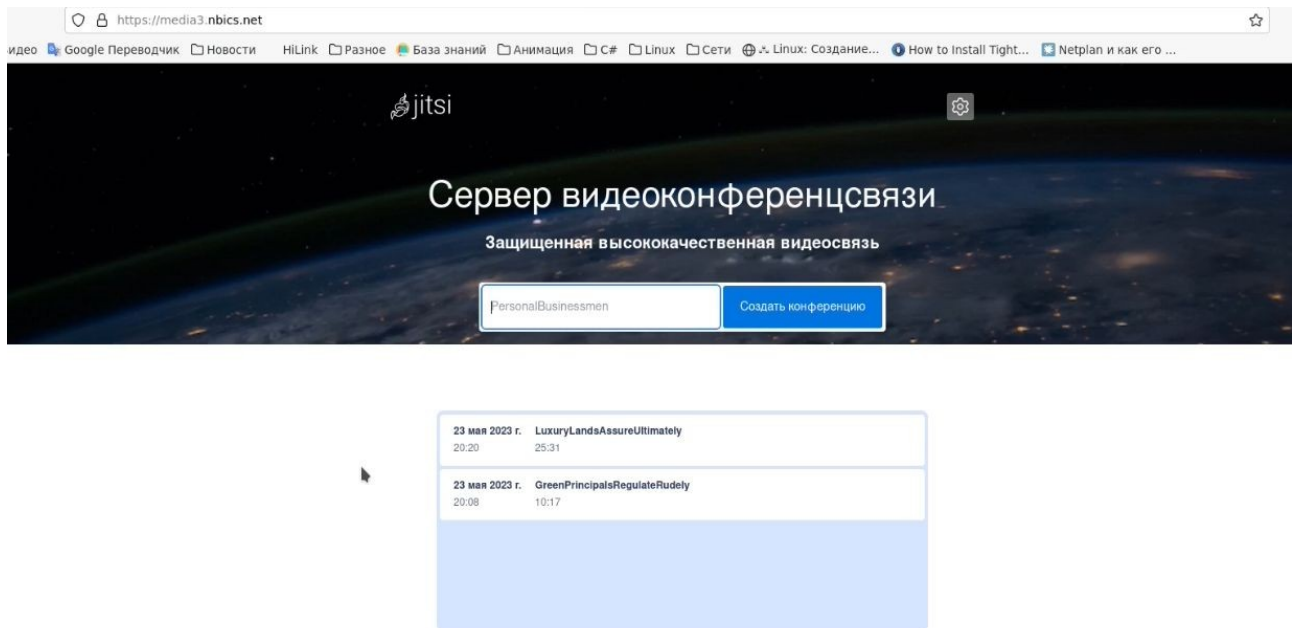


15. После установки Jitsi Meet проверяем статус основных служб:

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

Если все эти службы работают, значит всё нормально.

16. Запускаем сайт в браузере:



По умолчанию любой пользователь, без авторизации сможет войти, создать свою конференцию и приглашать туда людей.

Нам нужно настроить Jitsi так, чтобы только прописанный в системе пользователь мог авторизовываться на сайте и создавать конференции. Об этом — в следующей инструкции.