

# Лабораторная 6.

## Развертывание Asterisk

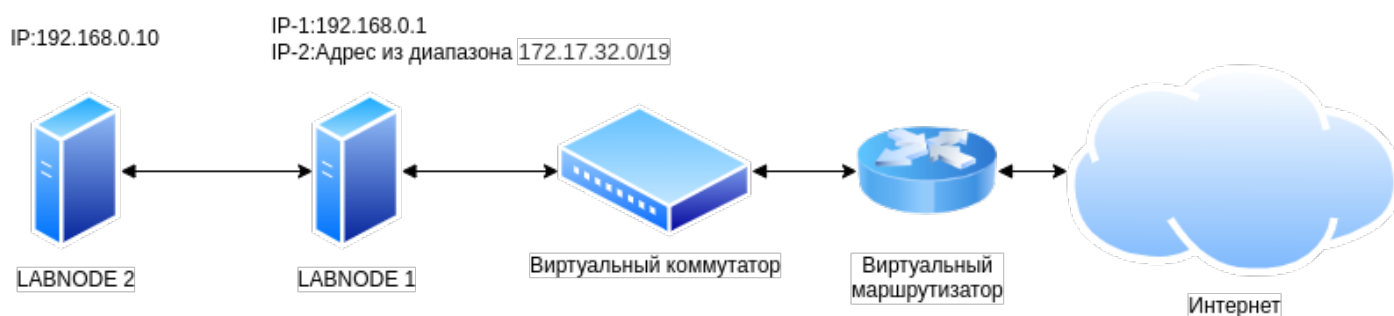
### Цель:

Получение базовых навыков работы в настройке программной АТС Asterisk.

### Задачи:

1. Подключится к облачной платформе СПбГУТ
2. Скачать исходные файлы Asterisk
3. Скомпилировать Asterisk
4. Настроить Asterisk
5. Запустить Asterisk
6. Зарегистрировать пользователя в программном телефоне Linphone

### Схема виртуального стенда:



### Ход работы:

#### Подготовка стенда

Для прохождения трафика с **Asterisk**, во внешнюю сеть нам необходимо настроить работу с внешней сетью, для этого нам нужно из узла **Labnode-2**, сделать маршрутизатор и маршрутизировать трафик, из сети **192.168.0.0/24** во внешнюю сеть.

Перейдите в проект **[group-name]-[team number]:dpi**

## Действия на инстансе **Labnode-1**

Для начала нам необходимо проверить включен ли модуль, ядра необходимый для перенаправления трафика

```
sysctl net.ipv4.ip_forward
```

**net.ipv4.ip\_forward = 0** - означает, что модуль ядра выключен

Для включения этого параметра необходимо ввести команду:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Проверяем, включился ли модуль пересылки пакетов:

```
cat /proc/sys/net/ipv4/ip_forward
```

Команда аналогична первой команде для проверки работы модуля, в данном случае вам должно быть выведено просто значение **1**

Введенные выше команды включают работу модуля, только на время пока работает ОС, и при перезагрузке будет сброшено для дефолтного состояния, чтобы это работало и после перезагрузки системы необходимо добавить в для включения этого параметра.

```
echo "net.ipv4.ip_forward = 1" | sudo tee -a /etc/sysctl.conf
```

Если мы хотим, чтобы правило дописанное нами применилось, стоит ввести команду

```
sudo sysctl -p
```

Выполняем настройку **iptables**

1. Добавим правило позволяющее пакетам, исходящим из сети 192.168.0.0/24 на интерфейсе eth0, проходить через фаервол без дополнительной проверки.

```
sudo iptables -A INPUT -i eth0 -s 192.168.0.0/24 -j ACCEPT
```

2. Добавим правило, которое разрешает проходить через фаервол пакетам, относящимся к уже установленным и работающим сетевым соединениям, таким как соединения по протоколам TCP или UDP.

```
sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

3. Добавим правило, которое разрешает прохождение всех пакетов, относящихся к уже установленному соединению или новому соединению, через интерфейс **eth0** к **eth1**

```
sudo iptables -A FORWARD -i eth0 -o eth1 -m conntrack --ctstate NEW,RELATED,ESTABLISHED -j ACCEPT
```

4. Добавим правило, которое разрешает прохождение всех пакетов, относящихся к уже установленному соединению или новому соединению, через интерфейс **eth1** к **eth0**

```
sudo iptables -A FORWARD -i eth1 -o eth0 -m conntrack --ctstate NEW,RELATED,ESTABLISHED -j ACCEPT
```

5. Добавляем правило для включения маскардинге для проходящих через интерфейс **eth1**

```
sudo iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j MASQUERADE
```

Возможна необходимость заменить название интерфейсов

Название интерфейса в примере	К какой сети относится адрес
eth1	172.17.32.0/19
eth0	192.168.0.0/24

## Действие на узле Asterisk

Установка Asterisk

Для прохождения трафика во внешнюю сеть на **labnode-2** необходимо прописать маршрут по умолчанию через **labnode-1**, для этого нам необходимо ввести команду:

```
sudo ip route add 0.0.0.0/0 via 192.168.0.1 dev eth0
```

Нам необходимо подготовить виртуальную машину **labnode-2**, для работы с s3, для этого необходимо выполнить несколько действий, сначала необходимо установить пакет для работы с s3

```
sudo apt install s3cmd -y
```

После этого нам необходимо скопировать файл конфигурации в домашнюю папку пользователя

```
cp /var/lib/cloud/s3cfg ~/.s3cfg
```

После этого у нас полноценно работает s3

Скачиваем Астерикс

```
s3cmd get s3://dpi/asterisk-certified-18.9-current.tar.gz
```

Распаковываем архив

```
tar xvf asterisk*
```

Переходим в директорию

```
cd asterisk-certified*/
```

Добавляем звуки и устанавливаем необходимые компоненты

```
sudo apt update
sudo apt -y install git curl wget libnewt-dev libssl-dev libncurses5-dev subversion libsqlite3-dev build-essential
libjansson-dev libxml2-dev uuid-dev
sudo contrib/scripts/get_mp3_source.sh
```

Устанавливаем зависимости Asterisk

```
sudo contrib/scripts/install_prereq install
```

Вам будет предложено выбрать [национальный код](#)

Собираем и устанавливаем Asterisk и заполняем конфигурациями по умолчанию для работы

```
sudo ./configure
sudo make menuselect
sudo make
sudo make config
sudo make install
sudo make progdocs
sudo make basic-pbx
sudo ldconfig
```

Добавляем новую группу **asterisk**

```
sudo groupadd asterisk
```

Создаем пользователя **asterisk**, и указываем ему домашнюю директорию **/var/lib/asterisk**

```
sudo useradd -r -d /var/lib/asterisk -g asterisk asterisk
```

Добавляем пользователя **asterisk** в группы **audio** и **dialout**

```
sudo usermod -aG audio,dialout asterisk
```

Меняем права и владельцев директорий которые мы используем в процессе работы

```
sudo chown -R asterisk:asterisk /etc/asterisk
sudo chown -R asterisk:asterisk /var/{lib,log,spool}/asterisk
sudo chown -R asterisk:asterisk /usr/lib/asterisk
sudo chmod -R 750 /var/{lib,log,run,spool}/asterisk /usr/lib/asterisk /etc/asterisk
```

Приводим **/etc/default/asterisk** к виду

```
AST_USER="asterisk"
AST_GROUP="asterisk"
COLOR=yes
```

Добавляем строки в **/etc/asterisk/asterisk.conf**

```
[options]
runuser = asterisk
rungroup = asterisk
```

Редактируем файл pjsip **/etc/asterisk/pjsip.conf** и приводим его к ввиду который указан ниже(удаляем существующую конфигурацию)

```
[transport-udp-nat]
type = transport
protocol = udp
bind = 0.0.0.0

[100]
type = endpoint
disallow = all
```

allow = alaw  
allow = ulaw  
direct\_media= no  
context = internal  
callerid = 100  
auth = 100-auth  
aors = 100

[100-auth]  
type = auth  
auth\_type = userpass  
username = 100  
password = 123

[100]  
type = aor  
max\_contacts = 5

[101]  
type = endpoint  
disallow = all  
allow = alaw  
allow = ulaw  
direct\_media= no  
context = internal  
callerid = 101  
auth = 101-auth  
aors = 101

[101-auth]  
type = auth  
auth\_type = userpass  
username = 101  
password = 123

[101]  
type = aor  
max\_contacts = 5

Файл **/etc/asterisk/extensions.conf**, приводим его к виду(удаляем существующую конфигурацию):

```
[globals]
[general]
[default]

[internal]
exten => _1xx,1,Dial(PJSIP/${EXTEN})
exten => s,n,Hangup
```

Запускаем Asterisk и добавляем его в автозагрузку

```
sudo systemctl restart asterisk.service
sudo systemctl enable asterisk.service --now
```

Проверка

```
sudo asterisk -rvvvv
```

```
pjsip show endpoints
```

```
asterisk*CLI> pjsip show endpoints

Endpoint: <Endpoint/CID.....> <State.....> <Channels.>
  I/OAuth: <AuthId/UserName.....>
  Aor: <Aor.....> <MaxContact>
  Contact: <Aor/ContactUri.....> <Hash.....> <Status> <RTT(ms)..>
Transport: <TransportId.....> <Type> <cos> <tos> <BindAddress.....>
Identify: <Identify/Endpoint.....>
  Match: <criteria.....>
  Channel: <ChannelId.....> <State.....> <Time.....>
  Exten: <DialedExten.....> CLCID: <ConnectedLineCID.....>
=====

Endpoint: 100/100                                     Unavailable  0 of inf
  InAuth: 100-auth/100
  Aor: 100                                             5

Endpoint: 101/101                                     Unavailable  0 of inf
  InAuth: 101-auth/101
  Aor: 101                                             5

Objects found: 2
```

```
pjsip show transports
```

```
transport transports
asterisk*CLI> pjsip show transports

Transport:  <TransportId.....>  <Type>  <cos>  <tos>  <BindAddress.....>
=====

Transport:  transport-udp-nat          udp          0          0  0.0.0.0:5060

Objects found: 1
```

---

Версия #41

Тарабанов Илья Федорович создал 1 сентября 2022 11:10:25

Тарабанов Илья Федорович обновил 14 декабря 2023 17:25:38