

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

SQUADUS

1.8

РУКОВОДСТВО ПО НАСТРОЙКЕ SIP

Версия 1

На 16 листах

Дата публикации: 18.03.2025

**Москва
2025**

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис», «MyOffice» и «Squadus» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

СОДЕРЖАНИЕ

1 Общие сведения	5
1.1 Назначение	5
1.2 О приложении	5
1.3 Системные требования	5
2 Настройка freeswith	6
2.1 Установка и настройка	6
2.2 Проверка работоспособности	7
3 Настройка Asterisk	9
3.1 Настройка и установка	10
3.2 Проверка работоспособности	15

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем документе применяют следующие сокращения с соответствующими расшифровками (табл. 1):

Таблица 1 — Сокращения и обозначения

Сокращение	Расшифровка
ОС	Операционная система
ПО	Программное обеспечение

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

Документ предназначен для настройки работы SIP шлюза Squadus, обеспечивающего возможность участия в конференциях с помощью телефона и звонков по номеру телефона.

1.2 О приложении

Squadus — приложение для рабочего общения с помощью текстовых, голосовых и видеосообщений, а также участия в конференциях в веб-браузерах и на операционных системах Windows, Linux, macOS.

Приложение Squadus входит в состав следующих продуктов:

- Squadus;
- Squadus PRO;
- «МойОфис Профессиональный 3».

Подробное описание возможностей приложения приведено в документе «Функциональные возможности»

1.3 Системные требования

Перечень требований к программному и аппаратному обеспечению Squadus приведен в документе «Системные требования».

2 НАСТРОЙКА FREESWITCH

2.1 Установка и настройка

Порядок настройки и установки freeswitch:

1. Добавить образ freeswitch на сервер с ролью operator.

Образ freeswitch не присутствует в дистрибутиве. Для добавления образа необходимо выполнить следующие команды:

```
docker load -i /srv/freeswitch-0.6.7.tar
docker tag hub.stageoffice.ru/freeswitch:0.6.7 squadus-sa-
1.installation.example.net:5060/freeswitch:0.6.7
docker push
squadus-sa-1.installation.example.net:5000/freeswitch:0.6.7
```

2. Добавить в `squadus_setup/extra_vars.yml` переменные, позволяющие установку freeswitch.

```
freeswitch_enabled: true
squadus_postgresql_enabled: true
```

Убедиться, что на сервере с ролью operator установлена Python-библиотека `psycopg2`.

3. Указать диапазон портов RTP.

Для настройки диапазона портов необходимо добавить в `squadus_setup/extra_vars.yml` переменные:

```
squadus_freeswitch_rtp_sip_min_port: 16384
squadus_freeswitch_rtp_sip_max_port: 16884
```

4. Добавить переменную окружения контейнеру `squadus_freeswitch`:

```
+++collections/ansible
collections/net/freeswitch/roles/freeswitch/tasks/sidecar_vars.yml
@@ -9,3 +9,4 @@
  SIDECAR_SSL_CHECK_CLIENT: "{{ freeswitch_sidecar_check_client_ssl | string |
lower }}"
  SIDECAR_SSL_KEY_PATH: "/etc/pki/tls/private/{{ tls_key_name }}"
  SIDECAR_USE_SSL: "{{ freeswitch_sidecar_tls_enabled | string | lower }}"
+ XML_RPC_PASSWORD: "{{ freeswitch_sidecar_xml_rpc_password }}"
```

5. Заменить пароли в файле `squadus_setup/main.yml`:

```
freeswitch_postgres_password: "oth9Qua2uufuquaingao"
iridium_freeswitch_sidecar_password: "aeruWeasShooGhofohpO"
iridium_mongodb_password: "Shijalgoo6lkOEitaita"
reeswitch_sidecar_xml_rpc_password: "sdDfiwerAWsdaitamert"
```

6. Указать доменное имя сервера `freeswitch` в файле `hosts.yml` в соответствующей ему группе `squadus_freeswitch`:

В файле `hosts.yml` уже созданы в примерах необходимые для работы с SIP группы:

- контейнер `iridium` — группа `squadus_apps`;
- контейнер `freeswitch` — группа `squadus_freeswitch`.

Пример:

```
squadus_freeswitch:  
  hosts:  
    squadus-sip-1.installation.example.net:
```

7. Выполнить установку с помощью команды:

```
ansible-playbook playbooks/squadus.yml --tags=iridium, freeswitch, squadus,  
caddy2, postgresql, otel_collector
```

8. Дальнейшую настройку выполнить с помощью панели Администрирование. Процесс настройки представлен в документе «Администрирование».

2.2 Проверка работоспособности

Для проверки работоспособности необходимо выполнить следующие действия:

1. Проверить, что в боковом меню появился значок 

При отсутствии значка, проверить что выполнены все операции в соответствии с документом «Администрирование».

2. Нажать на значок и открыть меню набора номера (рис. 1).

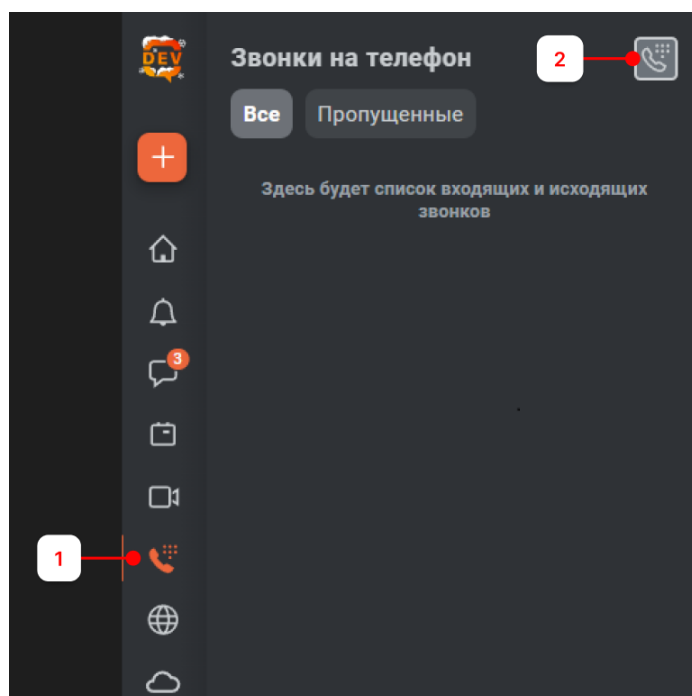


Рисунок 1 — Меню «Звонки на телефон»

3. В открывшемся окне введите номер 5000 и нажмите кнопку **Позвонить** (рис. 2).

Введите номер

×

Номер

✕

Чтобы ввести добавочный номер, нажмите и удерживайте * до появления запятой

1	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz
* ,	0 +	#

Позвонить

Рисунок 2 — Меню набора номера

3. Дождаться ответа от бота.

3 НАСТРОЙКА ASTERISK

Подключение цифровой автоматической телефонной станции к SIP-шлюзу продукта позволяет участвовать в конференциях с помощью телефона, поддерживающего тоновый набор номера.

Подключение к конференции осуществляется звонком с телефона на заранее настроенный номер на цифровой АТС (в этом примере используется номер «100»).

Чтобы попасть в конференцию, клиенту необходимо ввести ПИН (идентификатор) конференции, используя систему голосового меню IVR.

Приведенные конфигурационные файлы АТС Asterisk являются примерами и не предназначены для использования в production-среде, а также ориентированы на версию Asterisk 18.

На рисунке приведена схема взаимодействия продукта и АТС Asterisk (рис. 3).

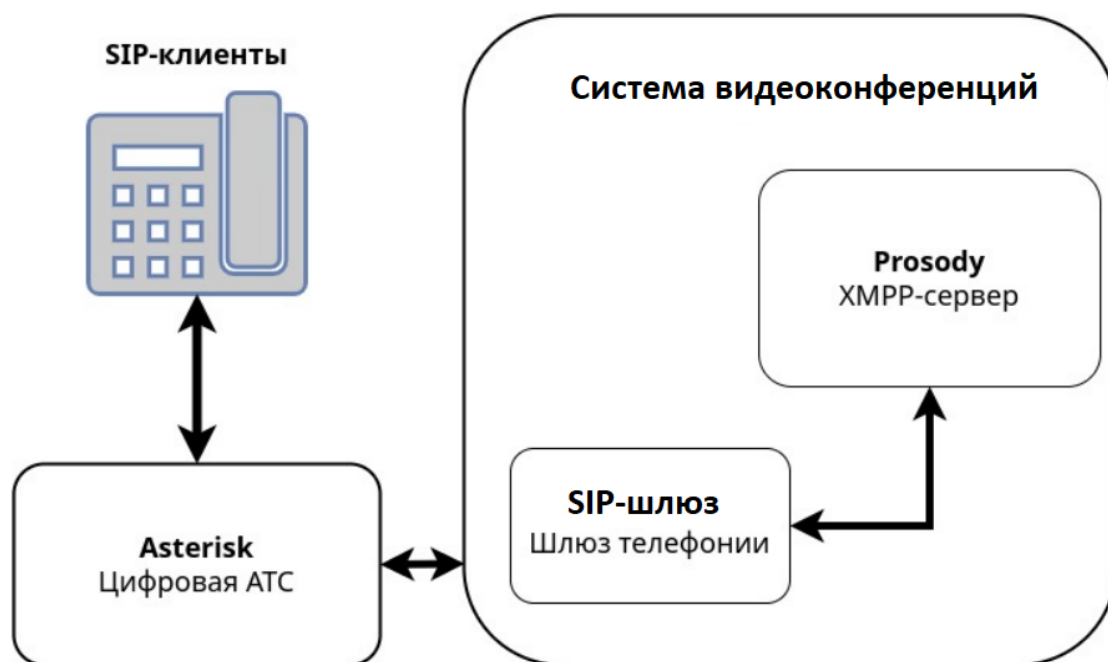


Рисунок 3 — Схема взаимодействия компонентов продукта с АТС Asterisk

3.1 Настройка и установка

Для настройки взаимодействия необходимо выполнить следующие действия:

1. Включить интеграцию SIP-шлюз для Jitsi, а также задать другие необходимые переменные в файле `group_vars/squadus_setup/extra_vars.yml`:

```
jitsi_sip_enabled: true
jigasi_sip_server: "10.0.0.1"
jigasi_sip_uri: "jigasi@{{ jigasi_sip_server }}"
jigasi_auth_password: "SECURE_PASSWORD_GOES_HERE"
jigasi_sip_password: "ANOTHER_SECURE_PASSWORD_GOES_HERE"
```

2. Настроить пользователя Asterisk с SIP номером 1001 в конфигурационном файле

```
pjsip.conf:

[transport-jigasi]
type = transport
protocol = udp
bind = 0.0.0.0:5160
external_media_address=10.0.0.1
external_signaling_address=10.0.0.1

[jigasi_auth]
type = auth
auth_type = userpass
password = {{ asterisk_jigasi_password }} # переменная использует значение
jigasi_auth_password
username = jigasi

[jigasi]
type = aor
max_contacts = 1
remove_existing = yes

[jigasi]
type = endpoint
transport = transport-jigasi
context = public
direct_media = no
disallow = all
allow = ulaw
allow = speex
auth = jigasi_auth
aors = jigasi
rtp_symmetric = yes
force_rport = yes
rewrite_contact = yes

{% for user in asterisk_users %}
[{{ user.name }}]
type = aor
max_contacts = 1

[{{ user.name }}]
type = auth
username = {{ user.name }}
password = {{ user.password }}
```

```
[{{ user.name }}]
type = endpoint
transport = transport-jigasi
context = {{ user.context }}
auth = {{ user.name }}
outbound_auth = {{ user.name }}
aors = {{ user.name }}
disallow = all
{%   for permission in user.allow %}
allow = {{ permission }}
{%   endfor %}

{% endfor %}[transport-jigasi]

type = transport
protocol = udp
bind = 0.0.0.0:5160

; внешний адрес хоста
external_media_address=10.0.0.1
external_signaling_address=10.0.0.1

[jigasi_auth]
type = auth
auth_type = userpass

; Пароль в незашифрованном виде. Такой же, как в переменной jigasi_auth_password
в файле extra_vars.yml
password = SECURE_PASSWORD_GOES_HERE
username = jigasi

[jigasi]
type = aor
max_contacts = 1
remove_existing = yes

[jigasi]
type = endpoint
transport = transport-jigasi
;context = jigasi-in

; public
context = public
direct_media = no
disallow = all
allow = speex
auth = jigasi_auth
aors = jigasi
rtp_symmetric=yes
force_rport=yes

rewrite_contact=yes

[1001]
type = aor
max_contacts = 1

[1001]
```

```
type = auth
username = 1001
password = 1234
[1001]
type = endpoint
transport = transport-jigasi
```

```
; public
context = public
auth = 1001
outbound_auth = 1001
aors = 1001
disallow = all
allow = g722
allow = speex
allow = alaw
```

3. Задать диапазон портов для RTP протокола в конфигурационном файле `rtp.conf`:

```
rtpstart=10050
rtpend=10100
```

4. Настроить интерактивное голосовое меню IVR в конфигурационном файле `extensions.conf`:

```
[public]
exten => 1001,1,Dial(PJSIP/1001,10)
exten => jigasi,1,Dial(PJSIP/jigasi,10)

exten => 100,1,Answer()
same => n,Set(CHANNEL(language)=ru)
same => n,Playback(greeting)
same => n,Set(Attempts=0)
same => n(getmeeting),Set(Attempts=${MATH(${Attempts}+1,i)})
same => n,Verbose(Attempts=: ${Attempts})
same => n,ExecIf(["${Attempts}" = "4"]?Gosub(Attempts-Error,s,1))
same => n,ExecIf(["${Attempts}" != "1"]?Playback(tryagain))
same => n,Read(PIN,beep,12)
same => n,Verbose(Result is: ${PIN})
same => n,GotoIf(["${PIN}" == "" ]?getmeeting)
same => n,Read(confpassword,enterpass&beep,6,,10)
same => n,Verbose(ResultPASS is: ${confpassword})
same => n,AGI(conferenceMapper.sh,${PIN})
same => n,Verbose(Result is: ${ROOM})
same => n,GotoIf(["${ROOM}" == "false" ]?invalidnum:joinmeeting)
same => n(invalidnum),Playback(simplewrongconfid)
same => n,Goto(getmeeting)
same => n(joinmeeting),Playback(thanks)
same => n,Playback(enterconf)
same => n,Verbose(Begin Result is: ${ROOM} -Pass: ${confpassword})
same => n,Dial(PJSIP/jigasi,,b(sub-headers^caller_handler^1(${ROOM},${confpassword})))
same => n,Verbose(0, Contacting Jitsi... Status is ${DIALSTATUS} );
same => n,Dial(PJSIP/jigasi,,b(sub-headers^caller_handler^1(${ROOM},${confpassword})))
same => n,Verbose(0, Contacting Jitsi... Status is ${DIALSTATUS} );
```

```
[easybell-in]
exten => _X.,1,Dial(PJSIP/jigasi,,b(handler^addheader^2))
exten => _X.,n,Playback(hello)
exten => _X.,n,Playback(conf-getconfno)
exten => _X.,n(getmeeting),Read(PIN,beep,12)
exten => _X.,n,Verbose(Result is: ${PIN})
exten => _X.,n,AGI(conferenceMapper.sh,${PIN})
exten => _X.,n,Verbose(Result is: ${ROOM})
exten => _X.,n,GotoIf([$[ "${ROOM}" == "false" ]?invalidnum:joinmeeting)
exten => _X.,n(invalidnum),Playback(conf-invalid)
exten => _X.,n,Goto(getmeeting)
exten => _X.,n(joinmeeting),Playback(conf-placeintoconf)
exten => _X.,n,Dial(PJSIP/jigasi,,b(sub-headers^caller_handler^1(${ROOM},${PIN})))

[sub-headers]
exten => caller_handler,1,NoOp(Set Header Jitsi-Conference-Room: ${ARG1} -Pass:
${ARG2})
same => n,Verbose(Result is: ${ARG1} -Pass: ${ARG2} )
same => n,Set(PJSIP_HEADER(add,Jitsi-Conference-Room)=${ARG1})
same => n,Set(PJSIP_HEADER(add,X-Room-Name)=${ARG1})
same => n,Set(PJSIP_HEADER(add,Jitsi-Conference-Room-Pass)=${ARG2})
same => n,Verbose(RETURN!!!!)
same => n,Return(${DIALSTATUS})
```

5. Создать AGI-скрипт, а также выставить необходимые права и сделать его исполняемым:

```
cat << 'EOF' > /var/lib/asterisk/agi-bin/conferenceMapper.sh
#!/bin/sh

if command -v curl 1>/dev/null; then
    HTTP_CLIENT="curl -s"
elif command -v wget 1>/dev/null; then
    HTTP_CLIENT="wget -q -O -"
else
    echo >&2 "ERROR! HTTP client is absent! Please Install curl or wget."
    exit 1
fi

jitsi_room=$(($HTTP_CLIENT
http://CHANGE_IT_TO_CONFERENCE_MAPPER_HOST:8001/conferenceMapper?id=$1 | sed -r
's#^.*conference": "(\\d+)@.*#\\1#g')

echo "SET VARIABLE ROOM \"${jitsi_room}\" "
EOF

chown asterisk:asterisk /var/lib/asterisk/agi-bin/conferenceMapper.sh

chmod 750 /var/lib/asterisk/agi-bin/conferenceMapper.sh
```

6. Настроить конфигурацию Caddy2. Для этого в файле `extra_vars.yml` добавить следующие пары ключ-значение:

```
caddy2_jitsi_sip_dialin_conference_mapper_backends: "{{ [groups['squadus_infra']  
[0]] }}"  
    caddy2_jitsi_sip_dialin_enabled: true  
    caddy2_jitsi_sip_dialin_phone_numbers:  
        message: "Phone numbers available."  
        numbers:  
            RU:  
                - "100"  
        numbersEnabled: true
```

7. Развернуть сервис `conferencemapper` на тех же хостах, что указаны в переменной `caddy2_jitsi_sip_dialin_conference_mapper_backends` в предыдущем пункте. При этом ожидается, что сервис будет доступен по TCP-порту 8001. В случае необходимости его можно переопределить переменной `caddy2_jitsi_sip_conference_mapper_port`.

8. Применить конфигурацию:

```
[root@squadus_infra~]# ansible-playbook playbooks/squadus.yml \  
--diff -t caddy2,jitsi
```

3.2 Проверка работоспособности

Для проверки работоспособности необходимо выполнить следующие действия:

1. Подключиться SIP-клиентом к хосту, на котором был развернут АТС Asterisk. Для аутентификации использовать логин и пароль, сконфигурированный ранее в файле `pjsip.conf`.
2. Создать конференцию Jitsi и найти идентификатор, он же ПИН см. пункт 3 (рис. 4).

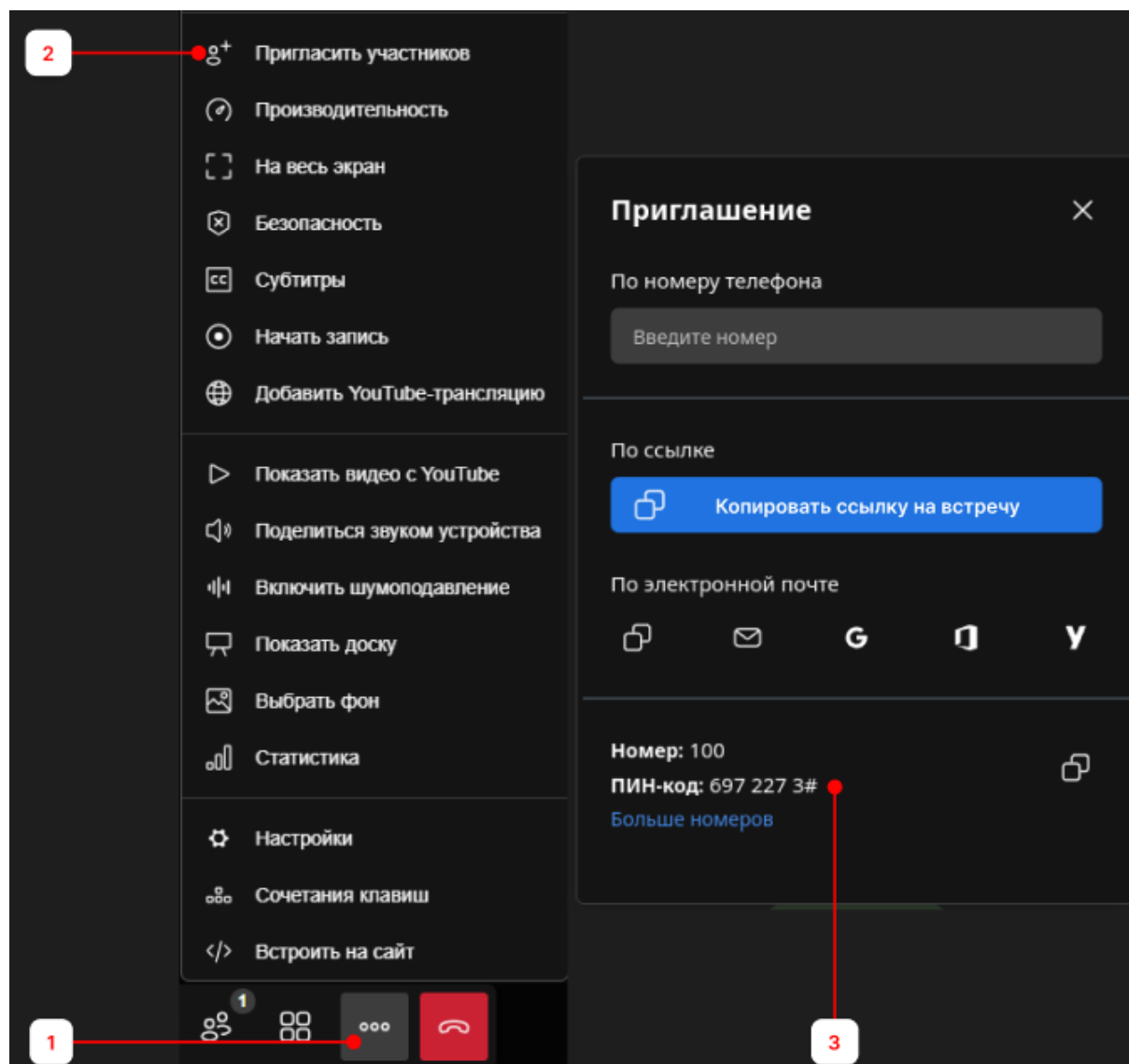


Рисунок 4 — Поиск идентификатора

3. Позвонить на сконфигурированный ранее в файле `extensions.conf` номер 100 SIP клиента.
4. Используя инструкции голосового меню, ввести идентификатор конференции. Так как конференция в данном случае не защищена паролем, в голосовом меню на этапе ввода пароля необходимо нажать решетку.

В результате к конференции должен присоединиться участник с именем 1001 (рис. 5).

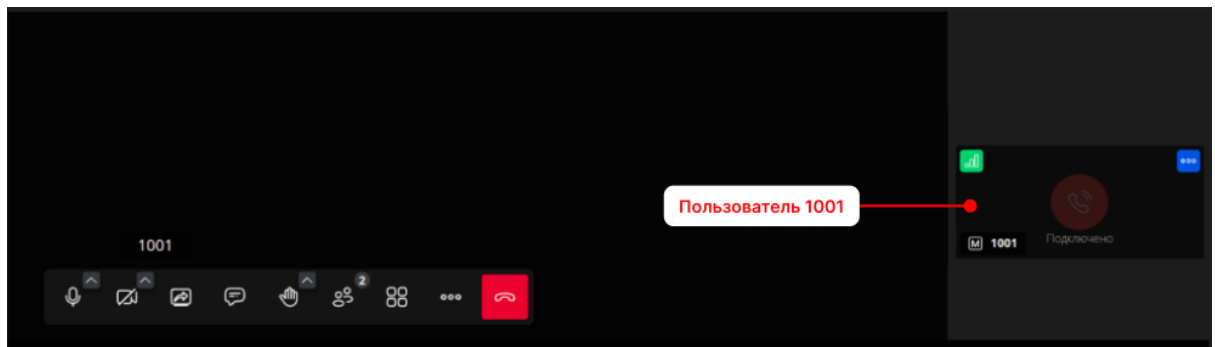


Рисунок 5 — Участник с именем 1001