

eXpress

Communication
System

Administrator's Guide

Media and Transcoding Servers

Build 3.36
03/17/2025



© Unlimited Production, 2025. All rights reserved.

All copyrights to the operating documentation are protected.

This document is included in the eXpress CS delivery package. It is subject to all terms and conditions of the license agreement. Neither this document, nor any part thereof, whether printed or electronic, may be copied or transmitted to third parties for commercial purposes without the express written permission of Unlimited Production.

The information contained in this document may be changed by the developer without special notice, which does not constitute a breach of obligations to the user by Unlimited Productions.

The server addresses, configuration file values, and user account data specified in the document are provided for example and are for informational purposes only. User data, including biometric data, are fictitious and do not contain personal data.

The provided components of eXpress CS as part of the delivery are intended exclusively for demonstration of functionality and are not intended for operation in a productive environment. For the correct functioning of the Express SC, before putting it into industrial operation, it is necessary to develop the architecture of a specific installation taking into account the specifics of the IT infrastructure.

Mailing address:	127030, Moscow, 24/1 Novoslobodskaya Street
Phone:	+7 (499) 288-01-22
E-mail:	sales@express.ms
Web:	https://express.ms/

TABLE OF CONTENTS

TABLE OF CONTENTS.....3

INTRODUCTION.....4

ARCHITECTURE5

SYSTEM REQUIREMENTS7

MOVING MEDIA COMPONENTS TO A SEPARATE SERVER8

Preparing for Launch8

Installing Media Server8

Setting Up Media Server connection to CTS.....10

Setting up Media Server.....12

 Setting Up JANUS, STUN and TURN Servers12

 Setting Up IP Telephony13

MOVING TRANSCODING TO A SEPARATE SERVER16

Setting Up Transcoding Server.....16

COMBINING FRONT, MEDIA AND TRANSCODING ROLES19

Setting Up a Server with combined Front, Media and Transcoding Roles19

INTRODUCTION

This manual is intended for administrators of the product eXpress Communication System (hereinafter referred to as eXpress CS, eXpress, system). This guide describes the process of moving eXpress media components to a separate server.

Attention! Starting with version 3.26, the architecture of the eXpress app has changed. To update the eXpress app to version 3.26 and above, move the eXpress media components to a separate Media Server as described in the document "Administrator's Guide."

If the eXpress media components remain deployed on the CTS Front (or Single) server, you need to set up the components in accordance with the section "Combining Front, Media and Transcoding Roles". This scheme is temporarily supported.

The release of updates compatible with the legacy architecture will be carried out until 15/02/2025. Updating the eXpress application to versions released after 02/15/2025 without changing the architecture may result in system inoperability.

The information provided in this document is relevant if the eXpress architecture does not contain Media and Transcoding servers that are externalized from the Single CTS or Front CTS server.

Product Support Service You can contact the product support service by e-mail support@express.ms. The page of the product support service on the Unlimited Production website is available at <https://express.ms/faq/>.

Website. Information on the product by Unlimited Production can be found on the website <https://express.ms/>.

The typical architecture assumes the availability of a separate Media server, as well as a separate Transcoding server.

The **Media server** is designed to organize video and audio communication between users (detailed information is provided in the document "Administrator's Guide. Volume 1. Installation", Media Server Installation section).

In general, the Media server should be placed in the demilitarized zone (DMZ), since publishing this server on the Internet is required.

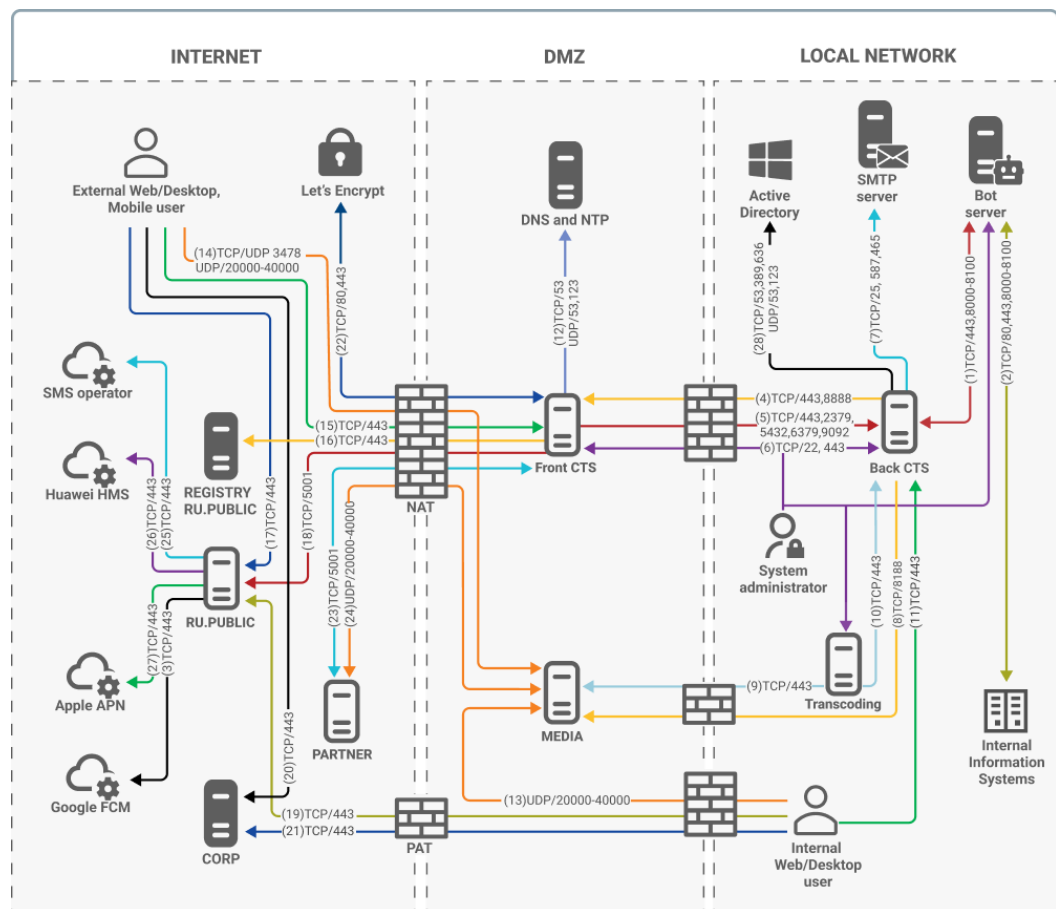
The **Transcoding server** is designed to transcode call and conference recordings into files (MP4 formats for video conferences and AAC for audio conferences).

In general, the Transcoding server must be located in the same network segment as the Back CTS components.

If the architecture uses a single corporate Single CTS server, the Transcoding server must be located in the same network segment as the Single CTS server.

If the architecture uses a detached corporate Front CTS and Back CTS server, then the Transcoding server must be located in the same network segment as the Back CTS server.

Figure 1. Typical deployment scheme of separated Media and Transcoding roles for Front+Back architecture



Information. For systems deployed on the Kubernetes platform, the Media role is moved to separate servers by default. The deployment of the Transcoding role to a separate server is similar to systems deployed in Docker.

Depending on the system deployment scheme, which is currently used, conversion to the target architecture can be performed in 1 or 2 steps (see [Figure 1](#)).

1. Moving eXpress media components from the corporate eXpress server (hereinafter referred to as the "CTS") to a separate Media server.
2. Moving the Transcoding container from the Media server to a separate Transcoding server.

The first step is mandatory for updating eXpress SC (for versions 3.26 and higher).

The second step ensures that the load on the Media server is reduced by separating the calling and transcoding services.

Attention! Information about network connections, which is provided in Table 1, takes precedence.

The eXpress media components consist of the following Docker containers:

- coturn (STUN/TURN service);
- janus (group call service);
- redis (KV storage);
- transcoding (responsible for recoding the recording into the output format).
- transcoding_storage (temporary storage service for recordings);
- nginx (web server that accepts connections from the outside and is responsible for routing internal connections);
- traefik (responsible for terminating TLS at the input).

After the first step is completed, the above containers are started on a separate Media server.

After completing the second step, the containers are distributed as follows:

- The Media server contains the following containers:
 - coturn (STUN/TURN service);
 - janus (group call service);
 - redis (KV storage);
 - transcoding_storage (temporary storage service for recordings);
 - nginx (web server that accepts connections from the outside and is responsible for routing internal connections);
 - traefik (responsible for terminating TLS at the input);
- The Transcoding server contains the transcoding container.

SYSTEM REQUIREMENTS

To determine the system requirements for the Media server and Transcoding server depending on the load, please refer to the document "Administrator's Guide. Volume 1. Installation", section "System Requirements".

MOVING MEDIA COMPONENTS TO A SEPARATE SERVER

Moving media components to a separate Media server involves several steps:

- [Preparing for Launch](#);
- [Installing Media Server](#);
- [Setting Up Media Server connection to CTS](#);
- [Setting up Media Server](#).

PREPARING FOR LAUNCH

1. Deploy the virtual machine(s) in the virtualization environment, taking into account the system requirements for the Media server (see the section "System requirements").
2. Set up network access (see [Table 1](#)) and NAT for the new Media server.

Table 1. Table of network interactions of the Media server

Direction	Source	Receiver	Port	Protocol	Purpose of the port
Incoming	Admin IP	Media	22	TCP	SSH
Incoming	CTS (Single/Back)	Media	8188	TCP	Management conference
Incoming	Any	Media	3478	TCP/UDP	TURN
Incoming	Any	Media	2000 0-4000 0	UDP	SRTP media
Incoming	Transcoding/Media	Media	443	TCP	Recording storage
Outgoing	Media	Any	Any	UDP	SRTP media
Outgoing	Media	DNS	53	TCP/UDP	DNS
Outgoing	Media	NTP server	123	UDP	NTP server
Outgoing	Media	registry.public.express	443	TCP	Docker registry
Outgoing	Transcoding/Media	CTS	443	TCP	Recording management

3. Assign a domain name to the Media server by entering it in the external and internal DNS, similar to split DNS for CTS.
4. Order SSL certificates for your assigned SSL domain name.
5. After issuing certificates, prepare the SSL certificate chain in PEM format and the unencrypted private key.

INSTALLING MEDIA SERVER

To install Media server:

1. Connect to the Media server via SSH.
2. Install the NTP server using the following command:

```
apt install chrony
```

If there are sources of exact time within the company, specify the NTP servers in the file `/etc/chrony/chrony.conf` as follows:

```
server ntp1.local  
server ntp2.local
```



```
server ntp3.local
```

Sample code:

```
systemctl enable chrony
systemctl restart chrony
```

To test connection to NTP servers, use the following command:

```
chronyc sources -v
```

3. Connect to the developer repository in Docker to download containers:

```
docker login -u Login -p Password registry.public.express
```

Note. Login and Password, which are issued by the developer, are used as login and password.

4. Download the installer container:

```
docker run --rm registry.public.express/dpl:voex-release dpl-
install | bash
```

A YAML file with containers and an installer will be downloaded from the repository to the server.

5. Create a working directory for the project:

```
mkdir -p /opt/express-voice
cd /opt/express-voice
echo DPL_IMAGE_TAG=voex-release > dpl.env
dpl --init
```

6. Install a certificate chain and SSL key for TURN and STUN servers.

```
mkdir -p certs
cp /somewhere/my-certificate-chain.crt certs/express.crt
cp /somewhere/my-unencrypted-key.key certs/express.key
```

7. Create a DH (Diffie Hellman) key:

```
openssl dhparam -out certs/dhparam.pem 2048
```

8. Open the /opt/express-voice/settings.yaml file for editing:

```
external_interface: eth0
janus_ws_enable: true
janus_ws_ip: 127.0.0.1
janus_wss_enable: false
janus_wss_ip: 127.0.0.1
- ' '
ccs_host: somehost.somedomain.sometld
phoenix_secret_key_base: *****
turnserver_server_name: localhost
turnserver_listening_ip: 127.0.0.1
api_internal_token: token
```

9. Make changes to the default settings (Table 2) and add the following parameter:

```
turnserver_external_ip:
- 1.2.3.4
```

10. Use the command below to generate the value of turnserver_shared_key:

```
cat /proc/sys/kernel/random/uuid | tr -d '-' | base64 | cut -
b 1-22
```

11. Add this parameter to the configuration:

```
turnserver_shared_key: YmNjY2VmNDk0ZTEwNDgzNj
```

12. If call recording is to be used, add the parameter (Table 2):

```
transcoding_storage_enabled: true
```

Table 2. List of Media Server Parameters

Setting name	Value
external_interface	Name of interface with external IP address
janus_keep_private_host	Enabling connection negotiation for all local IP addresses of the server
ccs_host	FQDN name of the Media server
api_internal_token	Token for API queries
janus_ws_acl	The addresses or networks of servers where the messaging container is located (for example, 172.18.0.)
janus_ws_ip - ip	An interface that uses janus websocket to manage conferences with the messaging service
janus_wss_enable janus secure websocket	Enabling janus secure websocket
janus_wss_ip	An interface that uses janus secure websocket
nat_1_1_mapping keep_private_host	When using NAT 1:1, the external IP address is specified and the private IP address saving mode is enabled
keep_private_host	The list of allowed IP addresses: • for one CTS server — its address: [1.2.3.4]; • if the CTS and Media servers are on the same server — empty list: []
phoenix_secret_key_base	Server key (leave unchanged)
turnserver_shared_key	Key to connect the Media server to the CTS server
turnserver_external_ip	External IP address
turnserver_listening_ip	External or internal IP address of the interface for TURN and STUN servers
transcoding_storage_enabled	Enable temporary storage of records service, disabled by default

13. Add the following parameters and set the parameter “janus_nat_1_1_mapping” equal to the value of the external IP address on the Internet from which port forwarding is performed:

```
janus_keep_private_host: true
janus_ws_ip: 172.17.0.1
janus_ws_acl: 172.19.0.
janus_nat_1_1_mapping: 1.2.3.4
```

14. Run the command to pre-generate configuration files:

```
dpl -p
```

15. Run the following command:

```
dpl -d
```

SETTING UP MEDIA SERVER CONNECTION TO CTS

To set up Media server connection to CTS:

1. Connect to the CTS (Single/Back) server via SSH.
2. Indicate the value of turnserver_shared_key that was generated on the Media server in /opt/express/settings.yaml. For example,

```
turnserver_shared_key: YmNjY2VmNDk0ZTEwNDgzNj
```

3. Remove the configuration files of the group calling service (janus) by running the following commands:

```
cd /opt/express-voice && dpl --dc down  
cd ~ && rm -rf /opt/express-voice
```

SETTING UP MEDIA SERVER

The procedure for setting up Media server includes:

- [Setting Up JANUS, STUN and TURN Servers](#) (mandatory setting);
- [Setting Up IP Telephony](#) (optional setting).

SETTING UP JANUS, STUN AND TURN SERVERS

To set up JANUS, STUN and TURN Servers:

1. Go to the following directory: `cd /opt/express-voice/`

```
cd /opt/express-voice
```

2. Start the Media server via the command line with the following command:

```
dpl -d
```

3. Open the administrator web interface.
4. In the "VoEx" section, to enable audio and video calls, add the new Media server names in the format `ws://internal_fqdn_media_cts:8188` for each server separately in the "Janus-Instances" (Figure 2) section. In the "Janus external host" section, enter the public IP of the media server.

Janus instances

Janus WS URL: * ws://janus.com:8188

Janus external host:

Janus internal host:

Interface: * ens192

Admin API URL: * https://janus.com:7088

Admin API secret: +

Enabled	Disabled by	Load	Janus WS URL	Interface	Janus external host	Janus internal host
☒		394	ws://janus.ru:8188	ens192	192.168.1.1	

Figure 2

5. Disable the old Janus server settings.
6. In the "TURN Server (comma separated)" field, enter the external FQDN of your server and the port number with a colon, for example, "express.firma.ru:3478";
7. In the "STUN Server (comma separated)" field, enter the external FQDN of your server and the port number with a colon, for example, "express.firma.ru:3478".
8. In the "Local VoEX network" field, specify the local network mask (see Figure 3).

VoEx

☒ Enable screen sharing for corporate users outside of contour

TURN Server (comma separated)

STUN Server (comma separated)

Voex local network

☐ Force relay ice

☒ Allow TCP ICE

☒ Enable audio streams mixing

☐ Enable VP9 video codec support

☐ Enable use of internal host for servers

List of servers which will use internal janus host (comma separated)

☒ Enable ability to record calls

Recording mode

Video and audio

Save

Figure 3

9. Apply the following marks, if necessary. Settings are described in Table 3:

Table 3. VoEx settings

Setting up	Description
Allow screen sharing outside of the closed contour	Allows the users to share their device screens with other users outside the CNT (RTS server users, trust server users, users who have left the CNT zone)
Use only Relay Ice candidates	Forced use of TURN server
Allow TCP ICE	The mark is set — TCP connection in TURN server is allowed. The mark is not set — TCP connection in TURN server is not allowed.
Enable audio stream mixing	Combines audio streams of calls directed from users to the server into one stream
Enable VP9 video codec	Item under development
Enable use of internal janus host for servers	Use the internal Janus host for the servers specified in the field below (see item 10)
Enable the ability to record calls	Allows users to record individual and group calls

Note. It is recommended to check “Allow screen sharing outside of the closed contour” and “Enable mixing of audio streams”.

10. In the “List of servers that will use the internal Janus host (comma separated)” field, enter the server with which communication will be conducted via the internal host.
11. Select the recording mode from the drop-down list.
12. Click the “Save” button.

SETTING UP IP TELEPHONY

To configure IP telephony:

1. In the “SIP” section, check the “SIP enabled” box (see Figure 4).

Figure 4. SIP settings window

2. Fill in the fields as follows (Table 4):

Table 4. SIP parameters

Field	Purpose
SIP server	ATE domain name or IP address (SIP trunk). If the port is other than UDP/5060, specify it separated by a colon
External address for Messaging SIP trunk	The field that adds the specified CTS server IP address to SIP INVITE messages. If the field is not filled, the default value is used: 127.0.0.1 Record format: sip:<IP or DNS name>:<port>
Host added to the user name when registering the SIP terminal	A field that is transmitted in the invite message towards the ATE. By default, the value ccs_host is added. If necessary, specify the host address from the configuration file
URI to connect to SIP Trunk	Back CTS address where the messaging container is installed. Filled out for Media and Back CTS deployments. Record format: sip:<IP or DNS name>:<port>
List of allowed SIP Trunk addresses	IP addresses from which calls will be received by the eXpress CS IP trunk. Specify at least two IP addresses: - ATE IP address; - address where the janus container is installed (SIP gateway, which is installed alongside with eXpress CS). All IPs or networks shall be indicated with a mask, for example – 10.10.10.1/32 for a single IP, 192.168.12.0/24 for a network. To deploy Single CTS, specify the IP address of the eXpress CS server itself (10.10.10.1/32) and the internal IP of the docker network interface (172.18.0.1/32) and the ATE. For Media and Back CTS deployment, specify IP Media and PBX
SIP Proxy	SIP telephony proxy server address or ATE address. SIP record format: <IP or DNS name>:<port>. It is not necessary to specify the port if it does not differ from standard UDP/5060

Prefix	A string inserted at the beginning of the dialed number when transmitting the number to the ATE and the number received from the ATE if the ATE sends the number without a prefix
PCRE template for prefix substitution	A regular expression to match the structure of the number to which a prefix will be inserted when making an outgoing call from eXpress CS. To prevent the prefix from being added to numbers, enter the expression <code>^[0-9](1)</code>
Preferred phone type	The type of phone from which calls will be made. Options: • phone; • IP phone; • phone (other); • IP-phone (other). The mapping of user object settings to these phone types is configured in the Active Directory section of the administrator web interface. The selected phone type will be hidden in the server user profiles when SIP integration is disabled.

3. Click the "Save" button.

Next, configure SIP trunk of the client ATE.

Attention! For all deployment schemes, it is mandatory to disable SIP trunk status checking.

For correct operation with the Single CTS deployment scheme, it is necessary to configure ATE 2 SIP trunk:

1. For calls from ATE to the System, specify the Single CTS destination IP;
2. For calls from the System to the ATE, specify the Media destination IP.

For correct operation with Front CTS and Back CTS deployment scheme, it is necessary to configure ATE 2 SIP trunk:

3. For calls from ATE to the System, specify the Back CTS destination IP;
4. For calls from the System to the ATE, specify the Media destination IP.

MOVING TRANSCODING TO A SEPARATE SERVER

SETTING UP TRANSCODING SERVER

To set up Transcoding server:

1. Connect to the dedicated server via SSH.
2. Create a folder for transcoding to work with:

```
mkdir -p /opt/transcoding
```

3. Install the Docker service:

```
curl -fsSL http://get.docker.com -o get-docker.sh && sh get-docker.sh
```

4. Specify Docker log storage parameters in the /etc/docker/daemon.json directory as follows:

```
{
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  }
}
```

5. Restart the Docker service:

```
systemctl restart docker
```

6. Go to the following directory: /opt/transcoding:

```
cd /opt/transcoding
```

7. Create a project variable:

```
echo "DPL_IMAGE_TAG=voex-release" > dpl.env
```

8. Connect to the developer's Docker repository to download containers.

```
docker login -u Login -p Password registry.public.express
```

Note. Login and Password, which are issued by the developer, are used as login and password.

9. Create and run a Docker container:

```
docker run --rm registry.public.express/dpl:voex-release dpl-install | bash
```

10. Initialize the VoEx project:

```
dpl --init
```

11. Open the file /opt/transcoding/settings.yaml in any text editor (for example, nano):

```
nano /opt/transcoding/settings.yaml
```

12. Disable the redis, coturn and janus services by setting them to false:

```
coturn_enabled: false
janus_enabled: false
redis_enabled: false
transcoding_storage_enabled: false
```


13. Add hosts for transcoding to work (see [Table 5](#)). The values can be copied from the Media Server from which the container is being transferred.

Important! Copy the `api_internal_token` values from the `/opt/express/settings.yaml` files located on the corresponding `ccs_hosts` servers. Copy the token values from the `/opt/express-voice/settings.yaml` files (`api_internal_token` value) located on the corresponding Media servers.

Example of transcoding host settings:

- for a single CTS server:

```
transcoding_hosts:
  cts:
    ccs_host: fqdn_cts
    api_internal_token: $api_internal_token_cts
    storages_tokens_mapping:
      fqdn_medial:
        token: $api_internal_token_media
        ssl_envs_prefix: "TSS"
# optional parameters
# if the certificates are not public, then disable
certificate verification:
tc-cts_env_override:
  TSS_SSL_ENABLED: true
  TSS_SSL_VERIFY: verify_none
```

- for multiple CTS servers:

```
transcoding_hosts:
  cts1:
    ccs_host: fqdn_cts1
    api_internal_token: $api_internal_token_cts1
    storages_tokens_mapping:
      fqdn_medial:
        token: $api_internal_token_medial
        ssl_envs_prefix: "TSS"
  cts2:
    ccs_host: fqdn_cts2
    api_internal_token: $api_internal_token_cts2
    storages_tokens_mapping:
      fqdn_media2:
        token: $api_internal_token_media2
        ssl_envs_prefix: "TSS"
# optional parameters
# if the certificates are not public, then disable
certificate verification:
tc-cts1_env_override:
  TSS_SSL_ENABLED: true
  TSS_SSL_VERIFY: verify_none
tc-cts2_env_override:
  TSS_SSL_ENABLED: true
  TSS_SSL_VERIFY: verify_none
```

Table 5. The list of transcoding server parameters

Setting name	Value
transcoding_hosts	The list of hosts objects (CTS) consists of the following parameters: • <code>cts</code> (<code>cts1</code>, <code>cts2</code>) — unique name, <code>fqdn_cts</code> can be used; • <code>ccs_host</code> — FQDN of the CTS server; • <code>api_internal_token</code> — token for API queries (copy from the <code>/opt/express/settings.yaml</code> files located on the

	corresponding <code>ccs_hosts</code> servers).
	It can contain multiple <code>cts</code> blocks if you have one transcoding server for multiple CTS servers
<code>storages_tokens_mapping</code>	The list of hosts objects consists of the following parameters: <code>fqdn_media</code> — FQDN of the Media server, must be unique; <code>token</code> — <code>api_internal_token</code> of the Media server; <code>api_internal_token_media</code> — API token specified on the Media server in the <code>file/opt/express-voice/settings.yaml</code> file in the <code>api_internal_token</code> parameter; <code>ssl_envs_prefix</code> — certificate prefix. May contain multiple <code>fqdn_media</code> blocks if the CTS server has more than one Janus(<code>janus_ws_url</code>)
<code>tc-ct_env_override</code>	Additional parameters for transcoding
<code>TSS_SSL_ENABLE</code>	Enable/disable advanced transcoding settings
<code>TSS_SSL_VERIFY</code>	Verification of certificate for transcoding

14. Start the service using the following command:

```
dpl -d
```

15. Check the status of containers using the following command:

```
docker ps -a
```

The command should result in transcoding containers appearing according to the value specified in the `transcoding_hosts` variable, for example:

```
root@express:/opt/express# docker ps -a | grep transcoding
fb0b3ba65346
registry.public.express/transcoding_manager:3.29.0
ca7d247a1552    registry.public.express/transcoding:3.29.1
db5801686026
registry.public.express/transcoding_storage:3.29.0
```

16. Check the availability of the Media Server using the following command:

```
curl https://fqdn-media/testtest
```

17. To get the Docker container logs on the Media server, run the following command:

```
docker logs voex-nginx-1 | grep testtest
```

The response must contain the request(s):

```
fqdn-media X.X.X.X - - [02/Oct/2024:08:50:34 +0000] "GET
/testtest HTTP/1.1" 204 0 "-" "curl/8.5.0" "X.X.X.X"
```

COMBINING FRONT, MEDIA AND TRANSCODING ROLES

In order to ensure a smooth transition to the new architecture, it is possible to temporarily place media components on one CTS Front (or Single) server.

Attention! Support for the architecture with combined Front, Media and Transcoding roles on one server will be discontinued on 15/02/2025. It is necessary to transfer Media and Transcoding roles to separate servers no later than the specified date. Updating the eXpress app to versions released after 15/02/2025 without changing the architecture may result in the system becoming inoperable.

To set up a co-located server for eXpress versions 3.26 and higher, you will need:

- determine the number of CPU cores used by the transcoding component;
- redefine the network port for interaction between media and transcoding components.

SETTING UP A SERVER WITH COMBINED FRONT, MEDIA AND TRANSCODING ROLES

To set up Transcoding server:

Connect to the shared Front server via SSH.

1. Go to the following directory: `cd /opt/express-voice/:`

```
cd /opt/express-voice
```

2. In the current configuration, which is located in the settings.yaml file, add the following lines:

```
transcoding_storage_enabled: true
traefik_bind_port_https: 8443
nginx_listen_http: false
ccs_host: fqdn_media
phoenix_secret_key_base: phoenix_secret_key
api_internal_token: api_internal_token_cts
coturn_options:
  cpuset: 10-11
  mem_limit: 2048m
janus_options:
  cpuset: 12-13
tc-cts_options:
  cpuset: 14-15
```

Where:

fqdn_media — FQDN address of the Front server

phoenix_secret_key — API token of the CTS server specified on the Back server in the /opt/express/settings.yaml file in the phoenix_secret_key_base parameter

api_internal_token_cts — API token of the CTS server specified on the Back server in the /opt/express/settings.yaml file in the api_internal_token parameter

cpuset — limitation of the use of CPU cores by container by core numbers, core numbering starts from 0

mem_limit — limitation of container memory usage

3. Change the block responsible for setting Transcoding according to the example below:

```
transcoding_hosts:
  cts:
    ccs_host: fqdn_cts
    api_internal_token: api_internal_token_cts
    storages_tokens_mapping:
      fqdn_media_or_ip:
        token: api_internal_token_media
        ssl_envs_prefix: "TSS"
        override_endpoint: https://fqdn_media:8443
  tc-cts_env_override:
    TSS_SSL_ENABLED: true
    TSS_SSL_VERIFY: verify_none
```

Where:

fqdn_cts — FQDN name of the CTS server specified on the Back server in the `/opt/express/settings.yaml` file in the `ccs_host` parameter

fqdn_media — FQDN address of the Front server

fqdn_media_or_ip — FQDN name or IP address of the Front server specified in the "VoEx" section, item "Janus instances"; the address must be specified in the same form (FQDN or IP)

api_internal_token_cts — API token of the CTS server specified on the Back server in the `/opt/express/settings.yaml` file in the `api_internal_token` parameter

api_internal_token_media — API token specified on the Media server in the file `/opt/express-voice/settings.yaml` in the `api_internal_token` parameter

4. Start the service using the following command:

```
dpl -d
```

5. Check the status of containers using the following command:

```
docker ps -a
```

The command should result in transcoding containers appearing according to the value specified in the `transcoding_hosts` variable, for example:

```
root@yc-msg-ext-voex-transcoding01:~# docker ps -a
CONTAINER ID   IMAGE                                CREATED        STATUS        PORTS
COMMAND
NAMES
dd5ca4e7bdee   registry.domain.ru/ccs/transcoding:3.29.0
"/bin/sh -c 'export ..." 45 hours ago  Up 22 hours
4000/tcp      voex-tc-cts-1
```