

v1.5_Installation manual

ELTEX Cloud Configuration Manager
ECCM Installation manual
Firmware version 1.5.0

Server Requirements

For stable operation, the server must meet the following requirements:

- Linux family operating system with docker support (Ubuntu server 20 is recommended and newer);
- RAM amount from 8 GB;
- CPU with virtualization support with at least 4 cores;
- free hard disk space from 40GB;
- 1Gbit/s network interface availability.

Obtaining sudo privileges without additional password input (optional)

On behalf of an unprivileged user, the following command must be run:

```
echo "$(whoami) ALL=(ALL) NOPASSWD: ALL" | sudo EDITOR='tee -a' visudo
```

Next, it is required to enter the password. After that, additional password entry is not required for executing `sudo` commands or switching to the superuser mode.

Installation Archive

The files required to run the project are distributed as a tar archive. They can be obtained from a public FTP server. Upload the archive to the server and unpack it. It is recommended to unpack the archive into the pre-created separate directory.

Commands example:

```
wget ftp://wifiguest:GuestWiFi@ftp.eltex.org/eccm/eccm-1.5.tar.gz
mkdir eccm
sudo tar -zxvf eccm-1.5.tar.gz eccm/
```

Docker and docker-compose installation

The easiest and fastest installation way is to go to the directory where the installation archive was unpacked and use the `compose-tools.sh` script using the `--install` flag:

```
cd eccm
sudo ./compose-tools.sh --install
```

To check the installed docker, the following commands can be run:

```
docker --version
```

To make sure that the command returns docker version no lower than the one specified in the example:

```
Docker version 20.10.8, build 3967b7
```

To check the installed docker-compose, the following commands can be run:

```
docker-compose --version
```

To make sure that the command returns a docker-compose version no lower than the one specified in the example:

```
docker-compose version 1.29.2, build 5becea4
```

ECCM start

The ECCM project is currently distributed as a set of files that allows all the necessary services using docker-compose. The project is divided into three docker stacks:

- database (PostgreSQL 12.5);
- authorization provider (OSS based on ORY Hydra and Oathkeeper);
- ECCM services stack.

This separation is conditioned by the need to ensure the horizontal project scaling and creating the possibility of integration with other projects of the company, for example, Eltex.EDM.

ECCM start script

Due to the abundance of the system startup parameters, a script was prepared that runs the project on a single host with performance parameters that allow servicing about 100 devices (actual performance depends on many factors, including hardware performance and the complexity of the network devices being serviced).

To start the project, go to the directory with the installation archive files and run the following command:

```
sudo ./compose-tools.sh --start <ECCM ADDRESS>
```

where ****ECCM ADDRESS**** is the IP address that will be used to connect to the server. For example, if the server address is 100.110.2.2, the command will look like the following:

```
sudo ./compose-tools.sh --start 100.110.2.2
```

It should be noted that the script applies the specified address to the containers being started, but does not rewrite it in files with variables (described in the next section). If using the `--start` key without specifying an address, then the addresses specified in the variable files will be used at startup.

Example of the console output at successful startup:

```
Starting postgres...
Creating network "postgres_default" with the default driver
Creating volume "postgres_postgres" with default driver
Creating postgres_postgres_1 ... done
Postgres started

set IDENTITY_PROVIDER_WEB_ADDRESS=100.110.2.2
set POSTGRES_HOST=100.110.2.2
Starting identity-provider...
WARNING: The JAEGER_AGENT_HOST variable is not set. Defaulting to a blank string.
Creating network "identity-provider_default" with the default driver
Creating volume "identity-provider_hydra-config" with default driver
Creating identity-provider_postgres-configurator_1 ... done
Creating identity-provider_hydra_1 ... done
Creating identity-provider_identity-provider_1 ... done
Creating identity-provider_nginx-proxy_1 ... done
Identity-provider started

set ECCM_WEB_ADDRESS=100.110.2.2
set IDENTITY_PROVIDER_HOST=100.110.2.2
```

```

set ECCM_BACKBONE_ADDRESS=100.110.2.2
set POSTGRES_HOST=100.110.2.2
Starting ECCM services...
WARNING: The DHCP_ENABLED variable is not set. Defaulting to a blank string.
Creating network "eccm_eltex-internal" with the default driver
Creating network "eccm_default" with the default driver
Creating volume "eccm_upgrader-data" with default driver
Creating volume "eccm_git-templates-data" with default driver
Creating volume "eccm_git-config-data" with default driver
Creating volume "eccm_monitoring-server-alertscripts" with default driver
Creating volume "eccm_monitoring-server-externalscripts" with default driver
Creating volume "eccm_monitoring-server-modules" with default driver
Creating volume "eccm_monitoring-server-enc" with default driver
Creating volume "eccm_monitoring-server-ssh_keys" with default driver
Creating volume "eccm_monitoring-server-ssl-certs" with default driver
Creating volume "eccm_monitoring-server-ssl-keys" with default driver
Creating volume "eccm_monitoring-server-ssl-ssl_ca" with default driver
Creating volume "eccm_monitoring-server-smmptraps" with default driver
Creating volume "eccm_monitoring-server-mibs" with default driver
Creating volume "eccm_monitoring-server-export" with default driver
Creating volume "eccm_monitoring-web-ssl-nginx" with default driver
Creating volume "eccm_rabbitmq-data" with default driver
Creating volume "eccm_oathkeeper-config" with default driver
Creating volume "eccm_hydra-configurator" with default driver
Creating volume "eccm_dhcp" with default driver
Creating eccm_oathkeeper-proxy_1          ... done
Creating eccm_hydra-configurator_1        ... done
Creating eccm_rabbitmq-service_1          ... done
Creating eccm_monitoring-service-worker_1 ... done
Creating eccm_dhcp_1                      ... done
Creating eccm_postgres-configurator_1     ... done
Creating eccm_monitoring-service_1        ... done
Creating eccm_postgres-monitoring-bouncer_1 ... done
Creating eccm_monitoring-server_1         ... done
Creating eccm_monitoring-web_1            ... done
Creating eccm_diff-checker_1              ... done
Creating eccm_upgrade-finish-checker_1    ... done
Creating eccm_monitoring-adapter_1        ... done
Creating eccm_data-presenter_1            ... done
Creating eccm_license-manager_1           ... done
Creating eccm_polemarch_1                 ... done
Creating eccm_map-manager_1               ... done
Creating eccm_ztp-manager_1               ... done
Creating eccm_backend-ui_1                ... done
Creating eccm_device-manager_1            ... done
Creating eccm_cron-manager_1              ... done
Creating eccm_communicator_1              ... done
Creating eccm_git-connector_1             ... done
Creating eccm_upgrader_1                  ... done
Creating eccm_web-gui_1                   ... done
ECCM started

```

To stop the project, run the command:

```
sudo ./compose-tools.sh --stop
```

To update images of all ECCM containers, run the command:

```
sudo ./compose-tools.sh --pull
```

To clean up all containers, volumes and ECCM networks, run the command:

```
sudo ./compose-tools.sh --clean
```

When running a script with the `--clean` flag, the data from the Postgres database will also be deleted. It is recommended to use this flag only when completely uninstalling ECCM from the server.

To start the project with the required logging level, run the command:

```
sudo ./compose-tools.sh --logging DEBUG
```

In 1.5 version has been added a DHCP server with Zero Touch Provisioning (ZTP) functionality support, which adds devices to the system automatically. Disabled by default, to activate, add the `--dhcp` key to the startup line:

```
sudo ./compose-tools.sh --start 100.110.2.2 --dhcp
```

Environment variables

The files required to run ECCM and containing environment and configuration variables are located in the directory where the installation archive was unpacked:

```
postgres/.env
postgres/data/postgresql.conf
identity-provider/.env
eccm/.env
```

postgres/.env

In `postgres/.env` there are variables that determine the behavior of the Postgres database when running it in a container. The table shows a list of these variables:

Variable	Default value	Description
COMPOSE_PROJECT_NAME	postgres	Name of the project in docker-compose (used for identification if several projects are running on the server)
POSTGRES_USER		Parameter that allows to override the default login for accessing the database
POSTGRES_PASSWORD		Parameter that allows to override the default password for accessing the database
POSTGRES_TIMEZONE	Asia /Novosibirsk	Time zone in which the system operates (specified in accordance with tz database, for example, Asia /Novosibirsk)
POSTGRES_SHM_SIZE	2gb	Allocated RAM limitation for working with the Postgres DB

postgres/data/postgresql.conf

The file contains parameters that affect the performance of the database. The default settings are sufficient for a test run and about 100 devices support. Instructions for the configuration changing can be found on the ECCM page on the <https://eltex-co.com/> website or contact ELTEX technical support.

identity-provider/.env

The `identity-provider/.env` file contains variables that define the behavior of a container group responsible for user authorization in the ECCM system.

```
# External address of nginx proxy, it is necessary to set the correct external address during installation
IDENTITY_PROVIDER_WEB_ADDRESS=<IP >
IDENTITY_PROVIDER_WEB_PORT=9090

## Postgres
# External address of the Postgres container, it is necessary to set the correct external address during
installation
POSTGRES_HOST=<IP >
POSTGRES_PORT=5432
POSTGRES_USER=tester
POSTGRES_PASSWORD=tester
```

The table shows a list of these variables:

Variable	Default value	Description
COMPOSE_PROJECT_NAME	eccm	Name of the project in docker-compose (used for identification if several projects are running on the server)
ECCM_PROFILE	production	
IDENTITY_PROVIDER_TAG	1.5	
IDENTITY_PROVIDER_REGISTRY	hub.eltex-co.ru	Docker registry address from which the system images will be retrieved. If using a local mirror, then its address can be specified
IDENTITY_PROVIDER_WEB_ADDRESS	192.168.0.1	The address where the identity-provider service will run
IDENTITY_PROVIDER_WEB_PORT	9090	Port for accessing the identity-provider service
POSTGRES_HOST	192.168.0.1	The address where the Postgres DB is running
POSTGRES_PORT	5432	Port for accessing the Postgres DB
POSTGRES_USER		Parameter that allows to override the default login for accessing the database
POSTGRES_PASSWORD		Parameter that allows to override the default password for accessing the database

eccm/.env

The `eccm/.env` file contains variables that define the behavior of the project.

```

COMPOSE_PROJECT_NAME=eccm

ECCM_PROFILE=production
ECCM_TAG=1.
5
ECCM_REGISTRY=hub.eltex-co.
ru

# Must be changed to real addresses during the installation
ECCM_BACKBONE_ADDRESS=192.168.0.1
ECCM_WEB_ADDRESS=192.168.0.1

# Identity provider
(external)

# Must be specified the real external address of identity-provider
IDENTITY_PROVIDER_HOST=192.168.0.1
IDENTITY_PROVIDER_PORT=9090
IDENTITY_PROVIDER_PATH=/identity-provider
IDENTITY_PROVIDER_HYDRA_PUBLIC_PATH=/hydra
/public

IDENTITY_PROVIDER_HYDRA_ADMIN_PATH=/hydra/admin

ECCM_WEB_PORT=80

ECCM_TIMEZONE=Asia/Novosibirsk
ECCM_LOGLEVEL=INFO

# Number of simultaneous operations performed with devices
# Cannot exceed the number of processor cores
MAX_PARALLEL_TASKS=4

```

The table shows a list of these variables:

Variable	Default value	Description
COMPOSE_PROJECT_NAME	eccm	Name of the project in docker-compose (used for identification if several projects are running on the server)
ECCM_PROFILE	production	
ECCM_TAG	1.5	
ECCM_REGISTRY	hub.eltex-co.ru	Docker registry address from which the system images will be retrieved. If using a local mirror, then its address can be specified
ECCM_BACKBONE_ADDRESS	192.168.0.1	The internal address where ECCM system will run with devices on the network
ECCM_WEB_ADDRESS	192.168.0.1	The address where the ECCM system web interface will run
IDENTITY_PROVIDER_HOST	192.168.0.1	The external address of identity-provider service
IDENTITY_PROVIDER_PORT	9090	Port for accessing the identity-provider service
ECCM_WEB_PORT	80	Port for accessing the web interface
ECCM_TIMEZONE	Asia /Novosibirsk	Time zone in which the system operates (specified in accordance with tz database, for example, Asia/Novosibirsk)
ECCM_LOGLEVEL	'INFO'	Project logging level
MAX_PARALLEL_TASKS	4	Number of simultaneous operations performed with devices
ECCM_MONITORING_SERVER_MEM_HARD_LIMIT	512M	Physical memory limit for docker-container

ECCM_MONITORING_WEB_MEM_HAR D_LIMIT	256M	Physical memory limit for docker-container
POSTGRES_HOST	192.168.0.1	The address where the Postgres DB is running
POSTGRES_PORT	5432	Port for accessing the Postgres DB
POSTGRES_USER		Parameter that allows to override the default login for accessing the database
POSTGRES_PASSWORD		Parameter that allows to override the default password for accessing the database

Access to the web interface

To connect to the ECCM web interface, enter the following address in the browser:

`http://<IP address of your server (ECCM_WEB_ADDRESS)>/`

Credentials used by default: login 'eccm', password 'eccm'.

Known issues and solution methods

Errors that occur when the project is started

Error:

```
ERROR: error while removing network: network eccm_eltex-internal id
324bd72dd9c107cf2ea48effb75d9e7ad2dfbc8f5f7317b89cd7f318d61d5c4b has active endpoints
```

Possible reason: docker daemon is not running. To check, run the command:

```
sudo systemctl status docker
```

If the status in the Active line is different from Active (running), then this is the problem.

Solution: start docker with the command:

```
sudo systemctl start docker
```

Error:

```
Got permission denied while trying to connect to the Docker daemon socket at [unix:///var/run
/docker.sock](unix://intdocs.eltex.loc/var/run/docker.sock): Get http://%2Fvar%2Frun%2Fdocker.sock
/v1.40/containers/json: dial unix /var/run/docker.sock: connect: permission denied
```

Possible reason: the start was made from an unprivileged user who was not added to the docker group.

Solutions:

Preferred: Add a user to the docker group. This can be done using the command:

```
sudo usermod -aG docker $(whoami)
```

Perform all operations with root privileges.

Error:

```
Services starting...
Creating network "eccm_eltex-internal" with the default driver
ERROR: Pool overlaps with other one on this address space
```

Possible reason: The subnet specified in ECCM_INTERNAL_SUBNETWORK is already being used by the docker.

Solution: a different subnet in the .env file must be selected. The subnets that already were created by docker can be viewed using the command:

```
sudo docker network inspect $(docker network ls --filter "DRIVER=bridge" --format '{{ .Name }}') -f '{{ .Name
}}' {{ (index .IPAM.Config 0).Subnet }}
```

Errors that occur when the project is stopped

Error:

```
ERROR: error while removing network: network eccm_eltex-internal id
324bd72dd9c107cf2ea48effb75d9e7ad2dfbc8f5f7317b89cd7f318d61d5c4b has active endpoints
```

Possible reason: docker has not completely cleared the cache.

Solution: restart docker with the command:

```
sudo systemctl restart docker
```