

v1.14_SoftWLC backup

- General information
- Installing and configuring keepalived
- Configuring rsync
 - Starting/stopping procedure
 - Configuring rsync server
 - Configuring synchronization launch
- Configuring MySQL replication
 - Data dump transfer and transfer to the second server
 - MySQL configuration
 - Creating user accounts
 - Starting replication
 - Starting replication on the second server
 - Starting replication on the first server
 - Checking replication from EMS-GUI
- Configuring MongoDB
 - replicaSet configuration
- Eltex-PCRF operation in cluster mode
 - Configuring PCRF cluster
- Configuring SoftWLC modules
- Changing configuration via GUI
 - Admin Panel
 - Portal Constructor
 - EMS-GUI

General information

SoftWLC controller backup is performed according to the master-slave model. Files critical for system operation (configuration files, firmware files, upload data files), MySQL database files (in master-master mode), MongoDB database files and DHCP server logs are synchronized. Such a model provides service availability and relevance of data on both controllers if one of them fails, network is inaccessible or power supply problems occur.

In configuration examples of this section, IP addresses will be referred to as <ip_server1>, <ip_server2> and <virtual_ip>, where:

- <ip_server1> — real ip address of the first server
- <ip_server2> — real ip address of the second server
- <virtual_ip> — virtual ip address

SoftWLC controllers backup configuration includes the following steps:

- installing and configuring keepalived
- configuring rsync
- Configuring MySQL replication
- configuring replicaSet MongoDB
- configuring Eltex-PCRF operation in cluster mode
- changing configuration of modules for them to use virtual IP

Installing and configuring keepalived

The main component for controller backup. Provides master role passing and system operation problems detecting. To install keepalived, contact the Wi-Fi Service center and get a relevant distribution kit.

Configuring rsync

In backup scheme, rsync manages synchronization of Eltex-EMS and Eltex-APB service files, and also firmware, configuration and AP configuration upload files. Rsync is a client-server software. Master server acts as a client and synchronizes slave server's directories with local ones.

Starting/stopping procedure

To enable rsync server, define the following value in the file `/etc/default/rsync`:

`RSYNC_ENABLE=true`

To start the service after stopping:

```
root@swlc01-server:/# service rsync start
```

To stop the service, the following command is used:

```
root@swlc01-server:/# service rsync stop
```

To check the service status, use the command:

```
root@swlc01-server:/# service rsync status
```

This will be followed by a message:

```
* rsync is running
```

if the service is running, or by

```
* rsync is not running
```

if it is not.

Configuring rsync server

The main configuration file of rsync server is located in `/etc/rsyncd.conf`. The listing is given below.

/etc/rsyncd.conf

```
[ems-conf]
path = /usr/lib/eltex-ems/conf/
use chroot = no
max connections = 2
lock file = /var/lock/rsyncd
read only = no
list = no
uid = root
auth users = backup
secrets file = /etc/rsyncd.secrets
strict modes = yes
# IP-address of the server that will access the resource, i.e., of the second server in the pair
hosts allow = <ip_server1> <virtual_ip>
ignore errors = no
ignore nonreadable = yes
transfer logging = no
timeout = 60
refuse options = checksum dry-run
dont compress = *.gz *.tgz *.zip *.z *.rpm *.deb *.iso *.bz2 *.tbz

[ems-tftp]
path = /tftpboot
use chroot = no
max connections = 2
lock file = /var/lock/rsyncd.tftp
read only = no
list = no
uid = root
auth users = backup
secrets file = /etc/rsyncd.secrets
strict modes = yes
hosts allow = <ip_server1> <virtual_ip>
ignore errors = no
ignore nonreadable = yes
transfer logging = no
timeout = 60
refuse options = checksum dry-run
dont compress = *.gz *.tgz *.zip *.z *.rpm *.deb *.iso *.bz2 *.tbz

[ems-wp]
path = /var/ems-data/WP
use chroot = no
max connections = 2
lock file = /var/lock/rsyncd.ems-wp
read only = no
list = no
uid = root
auth users = backup
secrets file = /etc/rsyncd.secrets
strict modes = yes
hosts allow = 10.62.8.121 10.62.8.122
ignore errors = no
ignore nonreadable = yes
transfer logging = no
timeout = 60
refuse options = checksum dry-run
dont compress = *.gz *.tgz *.zip *.z *.rpm *.deb *.iso *.bz2 *.tbz
```

hosts allow parameters are specified for master server. The following form is recommended:

```
hosts allow = <other_server_ip> <virtual ip>
```

To perform authentication, configure `*rsync*` user on both servers by creating files `/etc/rsyncd.secrets` on both servers and specifying login and password in them.

```
backup:rspasswd
```

Assign file access rights by running the following commands on both servers:

```
root@swlc01-server:/# chmod 600 /etc/rsyncd.secrets
```

Configuring synchronization launch

Create files `/etc/rsync_client.secrets`, specify the following password there:

```
root@swlc01-server:/# echo "rspasswd" > /etc/rsync_client.secrets && chmod 600 /etc/rsync_client.secrets
```

File synchronization is performed by **cron** task where `/usr/lib/eltex-ems/scripts/rsync_ems_backup.sh` is run. The script starts rsync client and synchronizes local directories with directories of the second (backup) server.

Synchronization can be started only if the server works in master state.

`/usr/lib/eltex-ems/scripts/rsync_ems_backup.sh`

```
#!/bin/bash

LOCKFILE="/run/lock/rsync_ems_backup"

# IP address backup server
HOST=<ip_server2>
# Check if we're root
if [ `whoami` != "root" ]
then
    echo "This script should be run by root."
    exit 1
fi

# Check and create lock file
if ! lockfile-create --use-pid -r 0 $LOCKFILE &> /dev/null ; then
    echo "Backup is already running"
    exit 0
fi

# Check - if we're master - try to perform backup to slave
SRVMODE=`cat /tmp/keep.mode`
if [ "$SRVMODE" == "MASTER" ]
then
    rsync -urlogtp --delete-after --password-file=/etc/rsync_client.secrets /usr/lib/eltex-ems/conf/
    backup@$HOST::ems-conf > /tmp/rsync_ems_conf.log 2>&1
    echo $? >> /tmp/rsync_ems_conf_result.log
    rsync -urlogtp --delete-after --password-file=/etc/rsync_client.secrets /tftpboot/ backup@$HOST::ems-
    tftp > /tmp/rsync_ems_tftpboot.log 2>&1
    echo $? >> /tmp/rsync_ems_tftpboot_result.log
    rsync -urlogtp --delete-after --password-file=/etc/rsync_client.secrets /var/ems-data/WP/ backup@$HOST::ems-
    wp > /tmp/rsync_ems_wp.log 2>&1
    echo $? >> /tmp/rsync_ems_wp_result.log
else
    echo "Not master. No action will be performed."
fi

lockfile-remove $LOCKFILE
```

where

- **backup** – login specified in `/etc/rsyncd.secrets` file
- **HOST** – another server's IP address

Create **cron** tasks on both servers to start synchronization every minute:

```
root@swlc01-server:/# crontab -l | { cat; echo "*/1 * * * * /usr/lib/eltex-ems/scripts/rsync_ems_backup.sh"; }  
| crontab
```



cron service should be started on both servers

Configuring MySQL replication

Backup of data stored in MySQL database is carried out by master-master replication. That means each server is both master and slave at the same time. The scheme implies writing all database updates of the first server to a special binary log. The second server reads the log and applies the changes. The second server replicates data from the first server, and vice versa (<http://dev.mysql.com/doc/refman/5.5/en/replication.html>). That allows having a relevant copy of a database on two hosts simultaneously. If connection fails, changes are accumulated and then synchronized after reconnection.

Data dump transfer and transfer to the second server

When configuring backup during operation (i.e. if the current server's MySQL already has data in it), it is necessary to replicate data to the second server. This can be done using the **mysqldump** utility.

Block the tables, take the dump, unblock the tables and copy the file to the second server:

```
root@swlc01-server:/# mysql -uroot -proot -e "FLUSH TABLES WITH READ LOCK;"  
root@swlc01-server:/# mysqldump -uroot -proot --databases ELTEX_PORTAL eltex_alert eltex_auth_service eltex_ems  
payments radius wireless > mysqldump_master.sql  
root@swlc01-server:/# mysql -uroot -proot -e "UNLOCK TABLES;"  
root@swlc01-server:/# scp mysqldump_master.sql <username>@<ip_server2>:/home/<username>/
```

Then generate a dump on the second server:

```
root@swlc01-server:/# mysql -uroot -proot < /home/<username>/mysqldump_master.sql
```

MySQL configuration

mysql daemon configuration aims at specifying binary logs writing parameters. The words *first server* and *second server* are conditional and used to refer to differences in server configurations.

Make the following changes in **[mysqld]** section of **/etc/mysql/my.cnf** configuration file:

Comment out or delete the following line on both servers:

```
bind-address = 127.0.0.1
```

Specify **server-id**. The servers should be given unique identifiers, e.g., for the first server:

```
server-id = 1
```

For the second server:

```
server-id = 2
```

Enable binary logs on both servers:

```
log_bin = /var/log/mysql/mysql-bin.log
```

specify **auto_increment_increment** (increment step) and **auto_increment_offset** (start point) parameters.

For the first server:

```
auto_increment_increment= 2
auto_increment_offset = 1
```

For the second server:

```
auto_increment_increment= 2
auto_increment_offset = 2
```

For both servers:

- specify databases for which logs will be written:

```
binlog-do-db = eltex_alert
binlog-do-db = eltex_ems
binlog-do-db = wireless
binlog-do-db = radius
binlog-do-db = eltex_auth_service
binlog-do-db = ELTEX_PORTAL
binlog-do-db = payments
```

- specify databases for which logs will not be written:

```
binlog-ignore-db = mysql
binlog-ignore-db = Syslog
binlog-ignore-db = performance_schema
binlog-ignore-db = information_schema
```

Restart **mysql** on each server and create a database for replication.

```
root@swlc01-server:/# service mysql restart
```

Creating user accounts

For replication to work, a service account should be created on both servers. The server will connect master server and get data changes using this account.

Create an account for replication on the first server:

```
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'replication'@'<ip_server2>' IDENTIFIED BY
'password';
FLUSH PRIVILEGES;
```

Create an account for replication on the second server:

```
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'replication'@'<ip_server1>' IDENTIFIED BY
'password';
FLUSH PRIVILEGES;
```



SELECT privilege is used to check replication performance from GUI EMS

Starting replication

Starting replication on the second server

Run the **show master status** command in MySQL console of the **first** server and analyze the values obtained:

```
mysql> show master status \G

***** 1. row *****
      File: mysql-bin.000001
      Position: 00000107
      Binlog_Do_DB: eltex_alert,eltex_ems,radius,wireless,eltex_auth_service,payments,ELTEX_PORTAL
      Binlog_Ignore_DB: mysql,Syslog,performance_schema,information_schema
      1 row in set (0.00 sec)
```

Remember the parameters **File** and **Position**.



It is recommended to set **Position** equal to 107. This is the position from which log file writing begins.

Configure and start the **second** server replication from the first one (**perform the following operations on the second server**):

```
mysql> STOP SLAVE;
mysql> CHANGE MASTER TO MASTER_HOST='<ip_server1>', MASTER_USER='replication', MASTER_PASSWORD='password',
MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=107;
mysql> START SLAVE;
```

where

- **MASTER_LOG_FILE='mysql-bin.000001'** – specify **File** value, got on the first server.
- **MASTER_LOG_POS=107** – specify **Position** value, got on the previous step (on the initial setting, 107 is recommended).

Check replication state on the second server:

```
mysql> show slave status \G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: <ip_server1>
      Master_User: replication
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000001
      Read_Master_Log_Pos: 107
      Relay_Log_File: mysqld-relay-bin.000001
      Relay_Log_Pos: 107
      Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB:
      Replicate_Do_Table:
      Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
      Replicate_Wild_Ignore_Table:
      Last_Errno: 0
      Last_Error:
      Skip_Counter: 0
      Exec_Master_Log_Pos: 107
      Relay_Log_Space: 107
      Until_Condition: None
      Until_Log_File:
      Until_Log_Pos: 0
      Master_SSL_Allowed: No
      Master_SSL_CA_File:
      Master_SSL_CA_Path:
      Master_SSL_Cert:
      Master_SSL_Cipher:
      Master_SSL_Key:
      Seconds_Behind_Master: 0
      Master_SSL_Verify_Server_Cert: No
      Last_IO_Errno: 0
      Last_IO_Error:
      Last_SQL_Errno: 0
      Last_SQL_Error:
      Replicate_Ignore_Server_Ids:
      Master_Server_Id: 2
1 row in set (0.00 sec)
```

If **Slave_IO_Running** and **Slave_SQL_Running** are set to «**Yes**», replication has been started successfully.

Starting replication on the first server

On the second server run:

```
show master status \G
mysql> show master status \G
***** 1. row *****
      File: mysql-bin.000001
      Position: 00000107
      Binlog_Do_DB: eltex_alert,eltex_ems,eltex_ont,radius,wireless,eltex_auth_service,payments,ELTEX_PORTAL
      Binlog_Ignore_DB: mysql,Syslog,performance_schema,information_schema
1 row in set (0.00 sec)
```

Configure and start first server replication from the second server (**run the following commands on the first server**):

```
CHANGE MASTER TO MASTER_HOST='<ip_server2>', MASTER_USER='replication', MASTER_PASSWORD='password',
MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=107;
START SLAVE;
```


Check replication state on the first server:

```
mysql> show slave status \G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: <ip_server2>
      Master_User: replication
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000001
      Read_Master_Log_Pos: 107
      Relay_Log_File: mysqld-relay-bin.000001
      Relay_Log_Pos: 107
      Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
...
```

the output given is incomplete, since other data is less important

If **Slave_IO_Running** and **Slave_SQL_Running** parameters are set to «Yes», **Master_Log_File** and **Read_Master_Log_Pos** values are given, replication is performed in both directions.

Checking replication from EMS-GUI

MySQL replication state can be controlled from GUI EMS. To do that, edit `/etc/eltex-ems/check-ems-replication.conf` configuration file. The changes should be made on both servers:

`/etc/eltex-ems/check-ems-replication.conf`

```
# Enable("Yes") / Disable("No") replication check
ENABLE_REPLICATION="Yes"

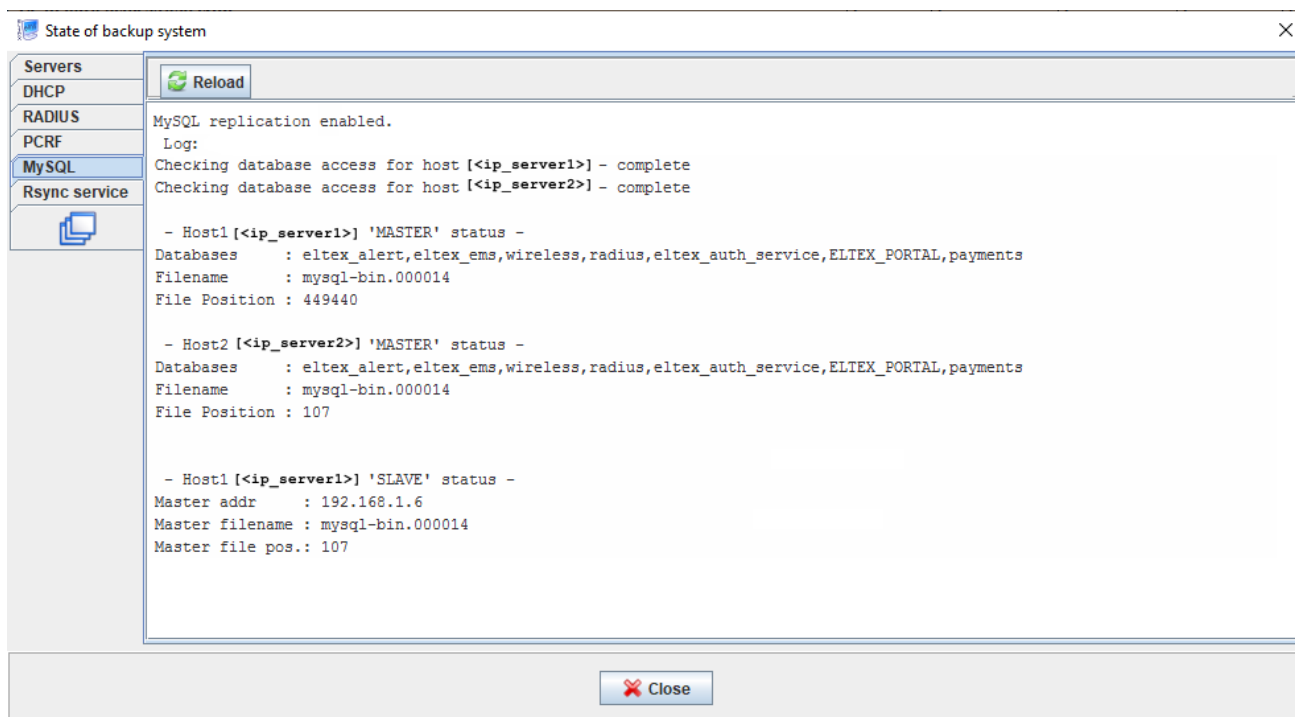
# The first replication host's address
HOST1=<ip_server1>
# The second replication host's address
HOST2=<ip_server2>

# mysql server access parameters
# mysql user
USER="replication"
# mysql password
PASSWORD="password"
```

where

- **ENABLE_REPLICATION** shows whether replication check is enabled (set to "Yes")
- **HOST1**, **HOST2** - servers' IP addresses
- **USER**, **PASSWORD** - user account login/password for replication.

After saving the changes, replication state can be checked via GUI EMS in *Information State of backup system MySQL*.




Results of replication state check on both servers and brief summary of check results are given in the section.

Configuring MongoDB

In MongoDB, replication is performed via grouping several (3 for standard configuration) nodes into Replica Set. Replica Set consists of one Primary node and several Secondary nodes (more information on <https://docs.mongodb.com/v2.4/administration/replica-sets>).

All data change operations are performed **only** on Primary. Thus, MongoDB automatically performs failover and replaces Primary with a working node if current Primary fails. But that requires 3+ nodes in Replica Set.

 In the default configuration, Replica Set that consists of two nodes completely goes down when a failure in one of them (even in Secondary).

replicaSet configuration

In `/etc/mongodb.conf` on both nodes:
add the line

```
replicaSet = <replica_set_name>
```

where `<replica_set_name>` is a name of replica set, which is chosen arbitrarily but should be the same for both servers.
Comment out or delete the line:

```
# bind_ip = 127.0.0.1
```

Restart MongoDB

```
root@swlc01-server:/# service mongodb restart
```

Open MongoDB console on the first node

```
root@swlc01-server:/# mongo
```

Create replica set configuration

```
> rs.initiate()
```

After a while, shell prompt should be changed to:

```
replica_set_name:PRIMARY>
```

If DNS is not used in the network, check if the first node was successfully added to Replica Set configuration.

```
replica_set_name:PRIMARY> rs.config()
{
  "_id" : "replica_set_name",
  "version" : 63243,
  "members" : [
    {
      "_id" : 0,
      "host" : "<hostname_server1>:27017"
    }
  ]
}
```



If the first element (at index 0) has hostname instead of IP address, and DNS is not used in the network, Replica Set configuration should be updated before adding the second node. If DNS is used, and DNS name is correct, the second node can be added.

To update the server's address, run the following commands:

```
replica_set_name:PRIMARY> conf = rs.conf()
replica_set_name:PRIMARY> conf.members[<>].host = "<ip_server1>:27017"
replica_set_name:PRIMARY> rs.reconfig(conf)
```

Check the current configuration again:

```
replica_set_name:PRIMARY> rs.config()
{
  "_id" : "relica_set_name",
  "version" : 63243,
  "members" : [
    {
      "_id" : 0,
      "host" : "<ip_server1>:27017"
    }
  ]
}
```

"host" parameter should contain the server's IP address.

Add the second node to Replica Set (run on the first server):

```
replica_set_name:PRIMARY> rs.add("<ip_server2>")
{ "ok" : 1 }
```

MongoDB error response can be generated if there is no connection to the second node (or bind_ip = 127.0.0.1 is specified there), or replSet is not configured. On the second node, MongoDB management console prompt should be changed to:

```
root@swlc01-server:/# mongo
replica_set_name:SECONDARY>
```

The same should be done for other nodes. Replica Set state can be checked by running the command `rs.status()` in MongoDB console.

Eltex-PCRF operation in cluster mode

Configuring PCRF cluster

Open 5701 tcp and 5801 tcp ports between PCRF servers

On servers, in `/etc/eltex-pcrf/hazelcast-cluster-network.xml` configuration files, specify network interfaces' addresses (lines 5 and 22 of the example contain the server's address, and lines 14 and 15 contain the list of all cluster members).

Configuration example:

```
<network>
  <!-- Write here public address of the node -->

  <!-- specify the server's own address here -->
  <public-address>ip_server1</public-address>
  <port auto-increment="false" port-count="100">5701</port>
  <outbound-ports>
    <ports>0</ports>
  </outbound-ports>
  <join>
    <multicast enabled="false"/>
    <tcp-ip enabled="true">
      <!-- List IP addresses of all cluster members (including this one) -->
      <member>ip_server1</member>
      <member>ip_server2</member>
    </tcp-ip>
    <discovery-strategies>
    </discovery-strategies>
  </join>
  <interfaces enabled="true">
    <!-- specify the server's own address here -->
    <interface>ip_server1</interface>
  </interfaces>
```

Permit cluster start in `/etc/eltex-pcrf/eltex-pcrf.json` configuration file:

```
"cluster.enable" : true,
```

Restart Eltex-PCRF using the command

```
root@swlc01-server:/# service eltex-pcrf restart
```

Cluster state check

- http://<ip_server1>:7070/cluster
- http://<ip_server2>:7070/cluster

```
{
  "data" : {
    "enabled" : true,
    "state" : "ACTIVE",
    "members" : [ {
      "address" : "ip_server1",
      "local" : true,
      "active" : true
    }, {
      "address" : "ip_server2",
      "local" : false,
      "active" : true
    } ],
    "messagesStats" : {
      "received" : 45157,
      "sent" : 45144
    },
    "mongo" : {
      "available" : false,
      "error" : "not running with --replSet"
    }
  },
  "key" : "PcrfErrorCode.success",
  "message" : "Success",
  "code" : 0,
  "args" : [ ]
}
```

Configuring SoftWLC modules

It is necessary to configure SoftWLC modules on both servers to interact with controller via virtual ip. The following configuration files should be changed:

/etc/eltex-apb/apb.properties

```
nbi.client.protocol=http
nbi.client.host=localhost
nbi.client.port=8080
nbi.client.file=/axis2/services/RadiusNbiService?wsdl
nbi.client.login=admin
nbi.client.password=password
nbi.client.pool.size=100

# Config filePath to cache
cache.config=/etc/eltex-apb/ehcache.xml

# list of permitted files to
hosts.file=/etc/eltex-apb/hosts.json

# RRM config
rrm.conf.file=/etc/eltex-apb/rrmconf.json

# Timeout waiting for subscribe-request after connecting the access point to the server, in seconds
subscribe.idle.timeout=60
```

- Change **localhost** to **<virtual_ip>** in line 2.

/etc/eltex-pcrf/eltex-pcrf.json

```
{
  "auth.address" : "0.0.0.0",
  "auth.port" : 31812,
  "auth.mac.open.timeout.s" : 3600,
  "auth.mac.welcome.service" : "WELCOME",
```

```

"acct.address" : "0.0.0.0",
"acct.port" : 31813,

"lease.saver.address" : "0.0.0.0",
"lease.saver.port" : 4381,

"aaa.instances" : 5,
"aaa.host" : "127.0.0.1",
"aaa.secret" : "testing123",
"aaa.auth.port" : 1812,
"aaa.acct.port" : 1813,
"aaa.rest.port" : 7080,
"aaa.timeout" : 10,
"aaa.attempts" : 1,

"web.monitoring.port" : 7070,

"cluster.enable" : false,
"cluster.eventBusPort" : 5801,

"radius" : {
  "url" : "jdbc:mysql://localhost/radius?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000&autoReconnect=true",
  "user" : "javauser",
  "password" : "javapassword",
  "max_pool_size" : 16
},

"mongo.pcrf" : {
  "connection_string": "mongodb://localhost:27017/pcrf",
  "db_name": "pcrf"
},

"session.storage" : {
  "session.check.period.s" : 300,
  "unauth.store.time.s" : 600,
  "interval.number.expired" : 3,
  "min.interval.s" : 45,
  "default.interval.s" : 600
},

"bras.coa" : {
  "coa.timeout" : 10,
  "coa.attempts" : 1,
  "remote.coa.port" : 3799,
  "executor.size" : 100,
  "log.clean.period.s" : 600,
  "log.store.period" : {
    "period" : 14,
    "unit" : "D"
  }
},

"sql.ems" : {
  "url" : "jdbc:mysql://localhost/eltex_ems?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000&autoReconnect=true",
  "user" : "javauser",
  "password" : "javapassword",
  "max_pool_size" : 16
},

"sql.wireless" : {
  "url" : "jdbc:mysql://localhost/wireless?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000&autoReconnect=true",
  "user" : "javauser",
  "password" : "javapassword",
  "max_pool_size" : 16
}
}

```

- Change `localhost` to `<virtualip>` in lines 28, 35, 61, 68.

/etc/eltex-portal-constructor/config.txt

```
# The number of attempts to log into Admin Panel before locking.
admin.attempts=5
# The number of minutes after reaching the maximum number of attempts to log into Admin Panel
admin.block.time=5
```

```
#####
#####Eltex Auth Service settings#####
#####
auth.service.address=localhost
auth.service.port=21812
auth.service.timeout=10
auth.service.secret=eltex
auth.service.retries=3
# pap, chap, mschapv2, eap-mschapv2, eap-md5, eap-ttls
auth.service.protocol=eap-md5
```

```
#####
#####DB ELTEX_PORTAL settings#####
#####
# SQL dialect corresponding to DB type
portal.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
# Use level 2 cache
portal.hibernate.use_second_level_cache=true
# use sql request caching
portal.hibernate.use_query_cache=true
# selecting neighbouring entities
portal.hibernate.batch_size=2
# the number of entries after the first iteration
portal.hibernate.fetch_size=100
# Show the generated sql
portal.hibernate.show_sql=false
#####
# DB driver
portal.db.driver=com.mysql.jdbc.Driver
# DB URL
portal.db.url=jdbc:mysql://localhost:3306/ELTEX_PORTAL?
max_allowed_packet=32362048&useUnicode=true&characterEncoding=utf8
# DB user's name
portal.db.username=javauser
# DB user's password
portal.db.password=javapassword
# request by pool to check connection validity
portal.db.validationQuery=select 1
# connection check period
portal.db.validationQueryTimeout=500
# a checkbox showing if a connection should be checked when being received from a pool
portal.db.testOnBorrow=true
# a checkbox showing if a connection should be checked when being returned to a pool
portal.db.testOnReturn=false
# a checkbox showing if a connection should be checked while idle
portal.db.testWhileIdle=true
# the number of ms at the borrow moment
portal.db.maxWait=10000
# minimum number of connections
portal.db.initialSize=1
# maximum number of connections
portal.db.maxActive=20
# maximum number of idle connections
portal.db.maxIdle=1
# minimum number of idle connections
portal.db.minIdle=1
# time between validity check cycles
portal.db.timeBetweenEvictionRunsMillis=10000
# minimum idle time
portal.db.minEvictableIdleTimeMillis=2000
# setting AutoCommit for connection
portal.db.defaultAutoCommit=false
```


Change `localhost` to `<virtualip>` in the lines 9, 36.

/etc/eltex-portal/config.txt

```
# DB ELTEX_PORTAL
# SQL dialect corresponding to DB type
portal.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
# Use level 2 cache
portal.hibernate.use_second_level_cache=true
# use sql request caching
portal.hibernate.use_query_cache=true
# selecting neighbouring entities
portal.hibernate.batch_size=1
# the number of entries after the first iteration
portal.hibernate.fetch_size=100
# Show the generated sql
portal.hibernate.show_sql=false

# DB driver
portal.db.driver=com.mysql.jdbc.Driver
# DB URL
portal.db.url=jdbc:mysql://localhost:3306/ELTEX_PORTAL?
max_allowed_packet=32362048&useUnicode=true&characterEncoding=utf8
# database user's name
portal.db.username=javauser
# database user's password
portal.db.password=javapassword
# request by pool to check connection validity
portal.db.validationQuery=select 1
# connection check period
portal.db.validationQueryTimeout=500
# a checkbox showing if a connection should be checked when being received from a pool
portal.db.testOnBorrow=true
# a checkbox showing if a connection should be checked when being returned to a pool
portal.db.testOnReturn=false
# a checkbox showing if a connection should be checked while idle
portal.db.testWhileIdle=true
# the number of ms at the borrow moment
portal.db.maxWait=10000
# minimum number of connections
portal.db.initialSize=5
# maximum number of connections
portal.db.maxActive=50
# maximum number of idle connections
portal.db.maxIdle=5
# minimum number of idle connections
portal.db.minIdle=1
# time between validity check cycles
portal.db.timeBetweenEvictionRunsMillis=10000
# minimum idle time
portal.db.minEvictableIdleTimeMillis=2000
# setting AutoCommit for connection
portal.db.defaultAutoCommit=false

portal.global_redirect_url=http://eltex-co.ru/
```

Change `localhost` to `<virtualip>` in line 18.

/etc/eltex-radius-nbi/radius_nbi_config.txt

```
# DB radius(alias=radius)
radius.jdbc.driver=org.gjt.mm.mysql.Driver
radius.jdbc.dbUrl=jdbc:mysql://localhost/radius?
zeroDateBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
```

```
radius.jdbc.username=javauser
radius.jdbc.password=javapassword
radius.jdbc.maxPoolSize=48
radius.jdbc.inUse=yes

# DB ems(alias=ems)
ems.jdbc.driver=org.gjt.mm.mysql.Driver
ems.jdbc.dbUrl=jdbc:mysql://localhost/eltex_ems?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000&noAccessToProcedureBodies=true
ems.jdbc.username=javauser
ems.jdbc.password=javapassword
ems.jdbc.maxPoolSize=48
ems.jdbc.inUse=yes

# DB wireless (alias=wireless)
wireless.jdbc.driver=org.gjt.mm.mysql.Driver
wireless.jdbc.dbUrl=jdbc:mysql://localhost/wireless?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
wireless.jdbc.username=javauser
wireless.jdbc.password=javapassword
wireless.jdbc.maxPoolSize=48
wireless.jdbc.inUse=yes

# DB logs (alias=logs)
logs.jdbc.driver=org.gjt.mm.mysql.Driver
logs.jdbc.dbUrl=jdbc:mysql://localhost/eltex_alert?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
logs.jdbc.username=javauser
logs.jdbc.password=javapassword
logs.jdbc.maxPoolSize=48
logs.jdbc.inUse=yes

# DB logs (alias=eltex_auth_service)
eltex_auth_service.jdbc.driver=org.gjt.mm.mysql.Driver
eltex_auth_service.jdbc.dbUrl=jdbc:mysql://localhost/eltex_auth_service?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
eltex_auth_service.jdbc.username=javauser
eltex_auth_service.jdbc.password=javapassword
eltex_auth_service.jdbc.maxPoolSize=48
eltex_auth_service.jdbc.inUse=no

# ems-northbound
ems.nbi.host=127.0.0.1
ems.nbi.port=8080
ems.nbi.path=northbound
ems.nbi.protocol=http

# eltex_auth_service
auth.port=22
auth.host=127.0.0.1
auth.username=username
auth.password=password

# freeradius-domain-1
freeradius-domain-1.port=22
freeradius-domain-1.host=192.168.0.1
freeradius-domain-1.username=username
freeradius-domain-1.password=password

# freeradius-domain-2
freeradius-domain-2.port=22
freeradius-domain-2.host=192.168.0.2
freeradius-domain-2.username=username
freeradius-domain-2.password=password

# tomcat url
tomcat.host=172.27.1.18
```

```

tomcat.port=8080

# pcrf stuff
pcrf.enabled=true
pcrf.url=http://localhost:7070
pcrf.username=admin
pcrf.password=password

# pcrf mongodb connector
pcrf.mongodb.enabled=true
pcrf.mongodb.uri=mongodb://localhost:27017/pcrf

# wifi-customer-cab mongodb connector
wificab.mongodb.enabled=true
wificab.mongodb.uri=mongodb://localhost:27017/wifi-customer-cab

# Eltex.SORM2.replicator MongoDB 'sorm2' connect
sorm2.mongodb.enabled=false
sorm2.mongodb.uri=mongodb://localhost:27017/sorm2

# Eltex.SORM2.replicator host to use API
sorm2.enabled=false
sorm2.url=http://localhost:7071
sorm2.username=admin
sorm2.password=password

#It enables records export to SORM3 while editing wifi users
sorm3.enabled=false

#####
#####DB ELTEX_PORTAL settings#####
#####
portal.db.driver=com.mysql.jdbc.Driver
portal.db.dialect=org.hibernate.dialect.MySQL5Dialect
portal.db.url=jdbc:mysql://localhost:3306/ELTEX_PORTAL?
max_allowed_packet=32362048&useUnicode=true&characterEncoding=utf8
portal.db.username=javauser
portal.db.password=javapassword
portal.hibernate.use_second_level_cache=false
portal.hibernate.use_query_cache=false

```

- Change `localhost` to `<virtualip>` in lines 3, 11, 19, 27, 35, 71, 77, 81, 85, 101
- Change `127.0.0.1` to `<virtualip>` in lines 49, 66

/etc/eltex-ngw/notification.properties

```
#Common gates settings
#Current gate used for each type (config name, for example smpp_gate.conf)
sms.gate.outgoing.sms.config=smsc_gate.conf
#For incoming sms/calls fields for numbers in configs should be set
sms.gate.incoming.sms.config=
sms.gate.incoming.call.config=

#Gate pool settings
sms.gate.pool.size=50
sms.gate.pool.wait.millis=5000

#Port to listen for requests
server.port=8040

#=====
#=====database settings=====
#=====
#mongodb.uri=mongodb://192.168.1.1,192.168.1.2:27017/notification-gw
mongodb.host=localhost
mongodb.port=27017
#mongodb.user=user
#mongodb.password=password
mongodb.name=notification-gw

#=====
#=====email settings=====
#=====
mail.smtp.submitter=test@email.com
mail.smtp.password=
mail.smtp.auth=true
mail.smtp.host=email.com
mail.smtp.port=587
mail.smtp.sendpartial=true
mail.smtp.starttls.enable=false
mail.smtp.connectiontimeout=5000
mail.gate.pool.size=20
mail.pool.wait.millis=10000
```

- Change `localhost` to <virtualip> in line 19

/etc/eltex-radius/local.conf

```
# Ports on which the server will listen
auth_port=1812
acct_port=1813
inner_tunnel_port=18121

# MySQL database
db_host="localhost"
db_port=3306
db_login="radius"
db_password="radpass"
db_name="radius"

# Enable checking that SSID is locked (1 - enabled, other values - disabled)
# Check is performed using query to 'wireless' database
ssid_check_enabled=1

# MySQL 'wireless' database
wireless_db_host="localhost"
wireless_db_port=3306
wireless_db_login="javauser"
wireless_db_password="javapassword"
wireless_db_name="wireless"

# PCRF
pcrf_host="127.0.0.1"
pcrf_port=7080
pcrf_enabled=1

# EAP
ca_cert_name="default.pem"
tls_key_password="eltex"

# Proxying
proxy_auth=0
proxy_domain_regex="^(.+\\.)?enterprise\\.root$"
proxy_host="127.0.0.1"
proxy_port=18121
proxy_secret="eltex"

# Ubiquiti vendor detection
ubi_vendor_regex="Apple|Ubiquiti"

# Settings of runtime NAS discovery
dynamic_clients=false
dynamic_client_subnet=192.168.0.0/16
dynamic_client_lifetime=3600
dynamic_client_rate_limit=false
```

- Change **localhost** to <virtualip> in lines 7, 18
- Change **127.0.0.1** to <virtualip> in line 25

/etc/eltex-wifi-cab/system.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="mongoaddress">mongodb://localhost:27017/wifi-customer-cab</entry>
  <entry key="nbiaddress">http://172.27.1.18:8080/axis2/services/RadiusNbiService?wsdl</entry>
  <entry key="nbi.serviceLogin.user">softwlc_service</entry>
  <entry key="nbi.serviceLogin.password">softwlc</entry>
  <!-- ,      eltex-auth-service -->
  <entry key="radius.auth.address">localhost</entry>
  <entry key="radius.auth.shareSecret">eltex</entry>
  <entry key="radius.auth.authPort">21812</entry>
  <entry key="radius.auth.acctPort">21813</entry>
  <entry key="radius.auth.timeout.sec">10</entry>
  <entry key="radius.auth.retries">5</entry>
</properties>
```

- Change localhost to <virtualip> in lines 4, 5, 9

/usr/lib/eltex-ems/conf/config.txt

```
# DB Event
poolName1=event
event.jdbc.driver=org.gjt.mm.mysql.Driver
event.jdbc.dbUrl=jdbc:mysql://localhost/eltex_alert?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
event.jdbc.username=javauser
event.jdbc.password=javapassword
event.jdbc.maxPoolSize=32
event.jdbc.inUse=yes
# remote db host access with su privileges
# event.ssh.login=
# event.ssh.password=
# event.ssh.port=

# DB Tree
poolName2=tree
tree.jdbc.driver=org.gjt.mm.mysql.Driver
tree.jdbc.dbUrl=jdbc:mysql://localhost/eltex_ems?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000&noAccessToProcedureBodies=true
tree.jdbc.username=javauser
tree.jdbc.password=javapassword
tree.jdbc.maxPoolSize=20
tree.jdbc.inUse=yes

# DB Ont
poolName3=ont
ont.jdbc.driver=org.gjt.mm.mysql.Driver
ont.jdbc.dbUrl=jdbc:mysql://localhost/eltex_ont?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
ont.jdbc.username=javauser
ont.jdbc.password=javapassword
ont.jdbc.maxPoolSize=40
ont.jdbc.inUse=yes

# DB Syslog
poolName4=syslog
syslog.jdbc.driver=org.gjt.mm.mysql.Driver
syslog.jdbc.dbUrl=jdbc:mysql://localhost/Syslog?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
syslog.jdbc.username=javauser
syslog.jdbc.password=javapassword
syslog.jdbc.maxPoolSize=4
syslog.jdbc.inUse=yes
# remote db host access with su privileges
# syslog.ssh.login=
```

```

# syslog.ssh.password=
# syslog.ssh.port=

# DB acsmain (alias=cpe)
poolName5=cpe
cpe.jdbc.driver=org.gjt.mm.mysql.Driver
cpe.jdbc.dbUrl=jdbc:mysql://localhost/acsmain?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
cpe.jdbc.username=javauser
cpe.jdbc.password=javapassword
cpe.jdbc.maxPoolSize=2
cpe.jdbc.inUse=yes

# DB acscmds(alias=cmds)
poolName6=cmds
cmds.jdbc.driver=org.gjt.mm.mysql.Driver
cmds.jdbc.dbUrl=jdbc:mysql://localhost/acscmds?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
cmds.jdbc.username=javauser
cmds.jdbc.password=javapassword
cmds.jdbc.maxPoolSize=2
cmds.jdbc.inUse=yes

# DB acsinf(alias=inf)
poolName7=inf
inf.jdbc.driver=org.gjt.mm.mysql.Driver
inf.jdbc.dbUrl=jdbc:mysql://localhost/acsinf?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
inf.jdbc.username=javauser
inf.jdbc.password=javapassword
inf.jdbc.maxPoolSize=2
inf.jdbc.inUse=yes

# DB acscache(alias=cache)
poolName8=cache
cache.jdbc.driver=org.gjt.mm.mysql.Driver
cache.jdbc.dbUrl=jdbc:mysql://localhost/acscache?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
cache.jdbc.username=javauser
cache.jdbc.password=javapassword
cache.jdbc.maxPoolSize=2
cache.jdbc.inUse=yes

# DB radius(alias=radius)
poolName9=radius
radius.jdbc.driver=org.gjt.mm.mysql.Driver
radius.jdbc.dbUrl=jdbc:mysql://localhost/radius?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
radius.jdbc.username=javauser
radius.jdbc.password=javapassword
radius.jdbc.maxPoolSize=40
radius.jdbc.inUse=yes
# remote db host access with su privileges
# radius.ssh.login=
# radius.ssh.password=
# radius.ssh.port=

# ----- SSID -----
# DB wireless (alias=wireless)
poolName10=wireless
wireless.jdbc.driver=org.gjt.mm.mysql.Driver
wireless.jdbc.dbUrl=jdbc:mysql://localhost/wireless?
useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=5000
wireless.jdbc.username=javauser
wireless.jdbc.password=javapassword
wireless.jdbc.maxPoolSize=30
wireless.jdbc.inUse=yes

```

Change localhost to <virtualip> in the lines 4, 17, 26, 35, 48, 57, 66, 75, 84, 98

Changing configuration via GUI

SoftWLC modules should also be configured via graphical interface.

Admin Panel

In the section *Settings Integration* replace localhost with a virtual IP address in *PCRF URL* and *URL NGW*-parameters:

Portal Constructor
Version: 1.15-128 (11.09.19 13:40:18)

English admin

System settings

Page title: Eitex WiFi Portal Administrator console

Icon (favicon): Change

Session timeout (min): 180

Portal host: localhost

Portal port: 9000

Uploading file size limit: 10 Mb

Uploading image resolution limit: 1440p (2560 x 1440)

Save

Portal Constructor

Replace localhost with a virtual IP address in the following sections:
System settings Portal Constructor

Portal Constructor
Version: 1.15-128 (11.09.19 13:40:18)

Dashboard

Gallery

System settings ▾

Portal Constructor

Alert settings

NBI access

NGW access

BRAS access

Payments DB

PCRF access

Mercury access

Wi-Fi Customer Cab

ESIA login

SATS integration


Volga platform access

Poll service access

Customer Cab

Portals


System settings

English  admin ▾

Page title

Eltex WiFi Portal Administrator console

Icon (favicon)

 Change

Session timeout (min)

180

Portal host

localhost

Portal port

9000

Uploading file size limit

10

Mb

Uploading image resolution limit

1440p (2560 x 1440) ▾

Save

System settings NBI access

Portal Constructor
Version: 1.15-128 (11.09.19 13:40:18)

Dashboard

Gallery

System settings ▾

Portal Constructor

Alert settings

NBI access

NGW access

BRAS access

Payments DB

PCRF access

Mercury access

Wi-Fi Customer Cab

ESIA login

SATS integration


Volga platform access

Poll service access

Customer Cab

Portals

NBI access

English  admin ▾

Protocol

HTTP ▾

Host

localhost

Port

8080


WSDL path

/axis2/services/RadiusNbiService?wsdl

Username

softwlc_service

Password

..... 

Save

System settings NGW access

Portal Constructor
Version: 1.15-128 (11.09.19 13:40:18)

Dashboard

Gallery

System settings ▾

Portal Constructor

Alert settings

NBI access

NGW access

BRAS access

Payments DB

PCRF access

Mercury access

Wi-Fi Customer Cab

ESIA login

SATS integration


Volga platform access

Poll service access

Customer Cab

Portals

NGW access

English  admin ▾

Hostlocalhost

Port8040

Save

System settings Payments DB

Portal Constructor
Version: 1.15-128 (11.09.19 13:40:18)

Dashboard

Gallery

System settings ▾

Portal Constructor

Alert settings

NBI access

NGW access

BRAS access

Payments DB

PCRF access

Mercury access

Wi-Fi Customer Cab

ESIA login

SATS integration


Volga platform access

Poll service access

Customer Cab

Portals

Payments DB

English  admin ▾

Use payment DB☒


Hostlocalhost

Database namepayments

Loginjavauser

Password

.....



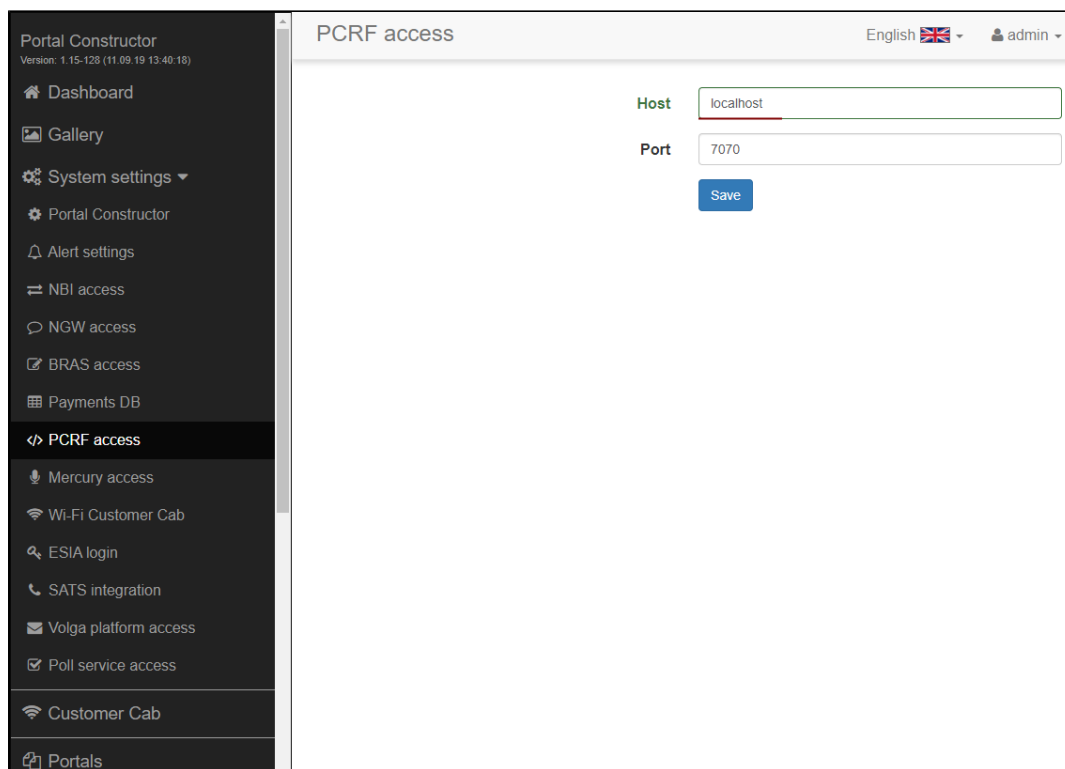
Connection pool size50

Cleanup timeout0 15 03 * * ?

Cleanup period0 15 10 L * * ?

Save

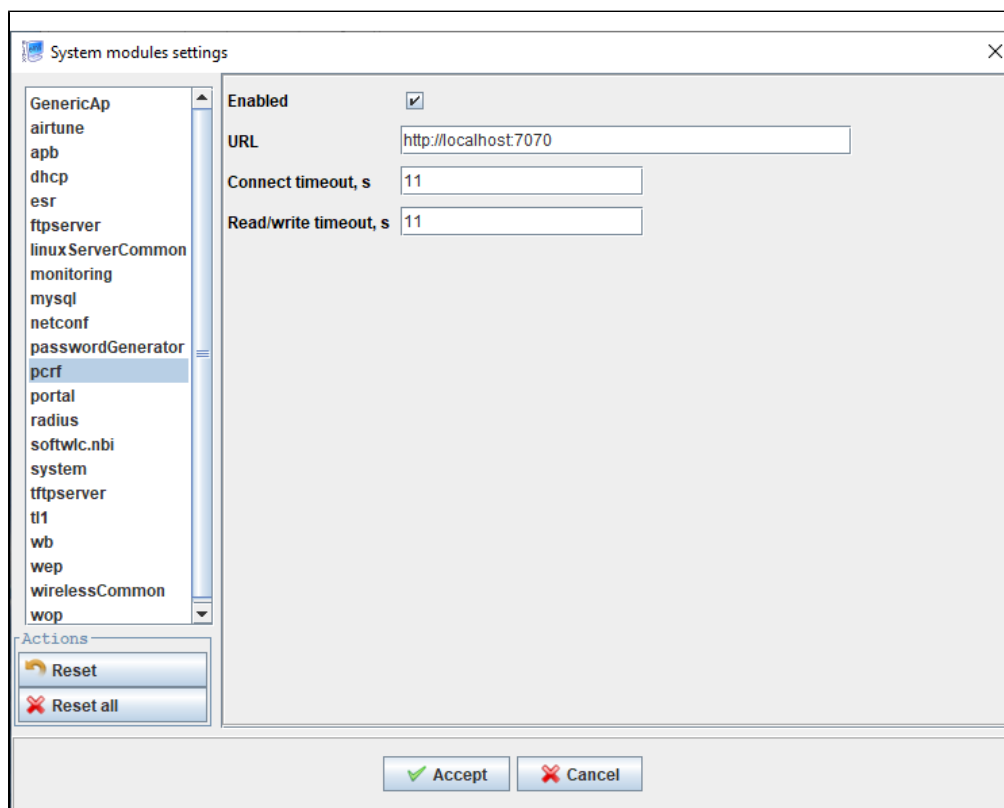
System settings PCRF access



EMS-GUI

In EMS GUI, replace localhost (or 127.0.0.1) with a virtual IP address in the following sections:

Administration EMS server configuration System modules settings pcrf



Administration EMS server configuration System modules settings radius

System modules settings

GenericAp
airtune
apb
dhcp
esr
ftpserver
linuxServerCommon
monitoring
mysql
netconf
passwordGenerator
pcrf
portal
radius
softwlc.nbi
system
tftpserver
ti1
wb
wep
wirelessCommon
wop

Actions
Reset
Reset all

RADIUS service name: eltex-radius

'Secret' for localhost: eltex

EMS host ip-address as RADIUS client: 127.0.0.1

Restart timeout: 45

Min. restart interval: 60

Accept Cancel

Administration EMS server configuration System modules settings softwlc.nbi

System modules settings

GenericAp
airtune
apb
dhcp
esr
ftpserver
linuxServerCommon
monitoring
mysql
netconf
passwordGenerator
pcrf
portal
radius
softwlc.nbi
system
tftpserver
ti1
wb
wep
wirelessCommon
wop

Actions
Reset
Reset all

Enabled: ☒

URL: http://localhost:8080/axis2/services/RadiusNbiService?wsdl

Username: softwlc_service

Password: softwlc

Connection timeout, s: 30

Request timeout, s: 120

Accept Cancel

Administration EMS server configuration System modules settings system

System modules settings

GenericAp
airtune
apb
dhcp
esr
ftpserver
linux ServerCommon
monitoring
mysql
netconf
passwordGenerator
pcrf
portal
radius
softwlc.nbi
system
tftpserver
tl1
wb
wep
wirelessCommon
wop

Language of GUI and system messages en_US

To forbid to send alerts to GUI ☐

EMS server IP address at management network 172.16.0.134

Temporary files directory /tmp/ems

Tomcat URL http://softwlc.eltex.loc:8080

Nested TOMCAT URL http://172.16.0.134:8080

Shared files directory /var/ems-data

Temporary DB files directory /tmp

Additional commands

Monitors log level ERROR

Send SNMP pre-request 'Check availability' ☒

Timeout for SNMP pre-request 'Check availability', ms 500

SNMP trap queue size 20000

Copy all alerts to Syslog ☐

Save auto-finish date to Events DB ☐

Asynchronous processes pool size 60

Asynchronous processes storing time, hours 24

ICMP availability poll enabled. ☒

SNMP availability poll enabled. ☒

Device response period (ICMP, SNMP), sec. 60

Ping utility flags for periodic device polling -c1 -i3 -t5 -p 500

Limit on the number of devices per 1 launch of fping ping utility 1000

Availability poll handlers pool size 10

ICMP, SNMP results storing time, sec 7200

Timeout for 'created' GUI session checking, sec 121

Accept Cancel

Administration EMS server configuration System modules settings tftpserver

System modules settings

GenericAp
airtune
apb
dhcp
esr
ftpserver
linux ServerCommon
monitoring
mysql
netconf
passwordGenerator
pcrf
portal
radius
softwlc.nbi
system
tftpserver
tl1
wb
wep
wirelessCommon
wop

TFTP server IP for station devices 172.16.0.134

Port (for embedded TFTP server) 69

Root directory for tftpd /tftpboot

Directory for FW files station_images

Directory for boot files uboot_images

Directory for configuration files ems

Reset

Reset all

Accept Cancel