v1.20_ SoftWLC

SoftWLC master-slave. (, ,), MySQL (master-master), MongoDB, DHCP . , , .

ip- <ip_server1>, <ip_server2> <virtual_ip>, :

- <ip_server1> - ip-
- <ip_server2> - ip-
- <virtual_ip> - ip-

SoftWLC :

- keepalived
- rsync
- MySQL
- replicaSet MongoDB
- Eltex-PCRF
- IP

keepalived

keepalived open source , (high availabilitty) (load-balancing). VRRP, Linux Vitrual Server (IPVS). Keepalived Eltex , . Keepalived SoftWLC, VRRP.

keepalived

(root):

```
root@master:/# apt update
root@master:/# apt install keepalived
```

Keepalived :

```
root@master:/# systemctl enable keepalived
root@master:/# systemctl start keepalived
```

keepalived

keepalived :

/etc/keepalived/keepalived.conf	
/etc/keepalived/check_ping.sh	EMS
/etc/keepalived/keep_notify.sh	, (MASTER, BACKUP, FAULT)
/etc/keepalived/mongo_switch.js	replicaSet MongoDB , VRRP

/etc/keepalived/keepalived.conf

```
! Configuration File for keepalived

global_defs {

   script_user root
   enable_script_security
}

vrrp_script check_network {
    script "/etc/keepalived/check_ping.sh"
    interval 5
    weight 50
    fall 3
    rise 3
    init_fail
    user root
}

vrrp_instance VI_SWLC {
    state BACKUP
    interface <interface>
    virtual_router_id 1
    track_script {
        check_network
    }
    track_interface {
        <interface> weight 50
    }
    priority 150
    advert_int 1
    nopreempt
        # Uncomment and comment "nopreempt" if preemption needed
        #preempt_delay 180
    authentication {
        auth_type PASS
        auth_pass eltex
    }
    virtual_ipaddress {
        <virtual_ip> dev eth0 label <interface>:1
    }

    notify_master "/etc/keepalived/keep_notify.sh master"
    notify_backup "/etc/keepalived/keep_notify.sh backup"
    notify_fault "/etc/keepalived/keep_notify.sh fault"

    unicast_peer {
        <ip_server1>
    }
}
```

- **<interface>** - ;
- **<virtual_ip>** - ip- ( );
- **<ip_server1>** - ip- ;

-

:

/etc/keepalived/check_ping.sh

```
#!/bin/bash

# host to ping
# there - default gw
HOST=<default_gw_ip>
# -q quiet
# -c nb of pings to perform
ping -q -c5 $HOST > /dev/null

# $? var keeping result of execution
# previous command
if [ $? -eq 0 ]
    then
        echo `date +"%T %F"` "OK gw reachable"
        EXIT_CODE=0
    else
        echo `date +"%T %F"` "ERROR gw unreacheble!"
        EXIT_CODE=1
fi

exit $EXIT_CODE
```

<default_gw_ip> - .

. , , SoftWLC .

keep_notify.sh

/etc/keepalived/keep_notify.sh

```
#!/bin/bash

MYSQL_USER="<mysql_user>"
MYSQL_PASSWORD="<mysql_password>"

mongo_set_role() {
    local role="$1"
    if [[ "$(which mongo)" ]]; then
        mongo --quiet --eval "var role=\"$role\"" admin /etc/keepalived/mongo_switch.js
        # Uncomment if using mongodb auth
        #mongo -u<username> -p<password> --quiet --eval "var role=\"$role\"" admin /etc/keepalived/mongo_switch.
js
    fi
}

if ! lockfile-create --use-pid -r 5 /tmp/keep.mode.lock; then
    echo "Unable to lock"
    echo "Unable to lock" > /tmp/keep.mode.lock.fail
    exit 0
fi

case "$1" in
    master)
    #  ems_reload_all
    echo "MASTER" > /tmp/keep.mode

    mongo_set_role master
    service eltex-ems restart
    service tomcat7 restart
    service eltex-ngw restart

    #   MySQL     -   ,
    #     heartbeat
    mysql -u$MYSQL_USER -p$MYSQL_PASSWORD -e "stop slave"
    mysql -u$MYSQL_USER -p$MYSQL_PASSWORD -e "start slave"
  ;;
 backup)
    echo "BACKUP" > /tmp/keep.mode
    mongo_set_role slave
    service mongodb restart
    service eltex-ems stop
    service tomcat7 stop
    service eltex-ngw stop
    mysql -u$MYSQL_USER -p$MYSQL_PASSWORD -e "stop slave"
    mysql -u$MYSQL_USER -p$MYSQL_PASSWORD -e "start slave"
 ;;
 fault)
    echo "FAULT" > /tmp/keep.mode
    mongo_set_role slave
    service mongodb restart
 ;;
 *)
    echo "Usage: $0 {master|backup|fault}"
    exit 1
esac

lockfile-remove /tmp/keep.mode.lock;

exit 0
```

<mysql_user> <mysql_password> - MySQL.

replicaSet MongoDB.

/etc/keepalived/mongo_switch.js

```
//
var role;

if (role != 'master' && role != 'slave') {
    throw "Role must be either master or slave";
}

var thisIsMaster = (role == 'master');
var status = rs.isMaster();
var thisHost = status.me;

print("Primary: " + status.ismaster + "; applying configuration ...");
var cfg = rs.conf();
for (var i = 0; i < cfg.members.length; i++) {
    var member = cfg.members[i];
    var self = (member.host == thisHost);
    if (self ^ thisIsMaster) {
        //   slave
        member.priority = 1;
        member.votes = 0;

        print(member.host + ": secondary");
    } else {
        //   master
        member.priority = 2;
        member.votes = 1;

        print(member.host + ": primary");
    }
}

var result = rs.reconfig(cfg, { force: !status.ismaster });
if (result.ok == 1) {
    print("Reconfiguration done");
} else {
    print(result);
}
```

, , :

```
root@swlc01-server:/# chmod +x /etc/keepalived/check_ping.sh
root@swlc01-server:/# chmod +x /etc/keepalived/keep_notify.sh
root@swlc01-server:/# chmod +x /etc/keepalived/mongo_switch.js
```

keepalived **/var/log/syslog** . , keepalived , -. **rsyslog**:

```
nano -w /etc/rsyslog.d/10-keepalived.conf
if $programname contains 'Keepalived' then /var/log/keepalived.log
if $programname contains 'Keepalived' then ~
```

rsyslog :

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

keepalived - **/var/log/keepalived.log** **/var/log/syslog** .

/ keepalived

:

```
service keepalived start
```

:

```
keepalived start/running, process 2471
```

:

```
root@master:/# service keepalived stop
```

:

```
keepalived stop/waiting
```

:

```
root@master:/# service keepalived status
```

:

```
keepalived start/running, process 2809
```

rsync

Rsync Eltex-EMS Eltex-APB, , , . Rsync . master- slave- .

rsync

rsync **/etc/default/rsync** :
RSYNC_ENABLE=true

/etc/ **rsyncd.conf** .

/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- ,      , ..
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 = <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
```

hosts allow master . :

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

rsync , **/etc/rsyncd.secrets**, .

```
backup:rspasswd
```

, :

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

/etc/rsync_client.secrets, :

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

cron, **/usr/lib/eltex-ems/scripts/rsync_ems_backup.sh.** rsync (backup) . , master.

/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
```

- **backup** – , **/etc/rsyncd.secrets**
- **HOST** - ip-

C **cron** :

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

```
root@swlc01-server:/# crontab -l
root@swlc01-server:/# */1 * * * * /usr/lib/eltex-ems/scripts/rsync_ems_backup.sh
```

-

```
root@swlc01-server:/# crontab -e

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /usr/bin/code
  4. /bin/ed

Choose 1-4 [1]: 1                                #
```

/

:

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

:

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

— , :

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

:

```
* rsync is running
```

,

```
* rsync is not running
```

.

MySQL

, MySQL, master-master (-). , master slave. , . , . (https://dev.mysql.com/doc/refman/5.7/en/replication.html). . , .

(MySQL) . **mysqldump**.

, , :

```
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_doors eltex_ems eltex_ngw 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>/
```

dump :

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

MySQL

mysqld . , .

[mysqld] /etc/mysql/mysql.conf.d/mysqld.cnf :

:

```
bind-address = 127.0.0.1
```

server-id. , , :

```
server-id = 1
```

:

```
server-id = 2
```

:

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

auto_increment_increment ( ) **auto_increment_offset** ( ).

:

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

:

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

:

- , :

```
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 = eltex_doors
binlog-do-db = eltex_ngw
```

- y, :

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

mysql .

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

. master .

MySQL :

```
GRANT SELECT, SUPER, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'replication'@'<ip_server2>' IDENTIFIED BY
'password';
GRANT SELECT, SUPER, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'replication'@'<ip_server1>' IDENTIFIED BY
'password'; #      EMS
FLUSH PRIVILEGES;
```

MySQL :

```
GRANT SELECT, SUPER, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'replication'@'<ip_server1>' IDENTIFIED BY
'password';
GRANT SELECT, SUPER, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'replication'@'<ip_server2>' IDENTIFIED BY
'password'; #      EMS
FLUSH PRIVILEGES;
```

SELECT GUI EMS

/usr/lib/eltex-ems/conf/config.txt, username / password (- javauser / javapassword)

```
GRANT ALL PRIVILEGES ON *.* TO 'username'@'%' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

MySQL **show master status** :

```
mysql> show master status \G

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

File Position.

Position 107. -.

( ):

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

,

- MASTER_LOG_FILE=**'mysql-bin.000001'** – **File**, .
- MASTER_LOG_POS=**107** – **Position**, (107).

:

```
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)
```

Slave_IO_Running Slave_SQL_Running «Yes», .

:

```
show master status \G
mysql> show master status \G

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

( ):

```
STOP SLAVE;
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;
```

:

```
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
...
```

,

Slave_IO_Running Slave_SQL_Running «Yes», **Master_Log_File Read_Master_Log_Pos** .

EMS-GUI

MySQL GUI EMS. **/etc/eltex-ems/check-ems-replication.conf**. .

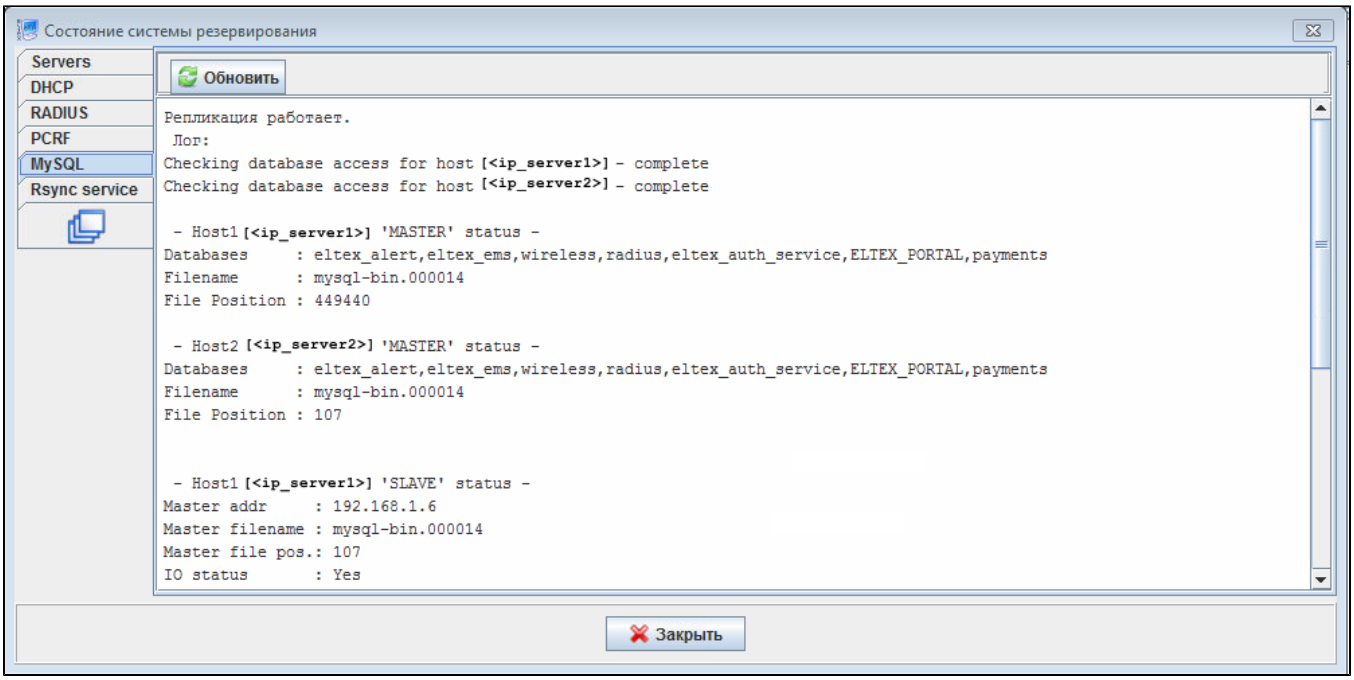
/etc/eltex-ems/check-ems-replication.conf

```
# ("Yes") / ("No")
ENABLE_REPLICATION="Yes"

#
HOST1=<ip_server1>
#
HOST2=<ip_server2>


#    mysql
#  mysql
USER="replication"
#  mysql
PASSWORD="password"
```

,

- **ENABLE_REPLICATION** - ("Yes")
- **HOST1, HOST2** - ip-
- **USER, PASSWORD** - / , .

, GUI EMS *MySQL.*



.

MongoDB

MongoDB (- 3) Replica Set. Replica Set primary secondary (https://docs.mongodb.com/v4.0/administration/replica-set-deployment/). :

Primary — mongoDB.

Secondary — () real-time .

Arbiter — , , .

mongo-db arbiter:

vCore: 1, 64-bit x86 CPUs

vRAM: 2

vHDD: 20

, MongoDB (SoftWLC 4.0.26).

primary. MongoDB failover primary , primary . 3+ replica set.

Replica Set (secondary).

replicaSet

`/etc/mongod.conf` :

/

```
replication:
   replSetName: "<replica_set_name>"
```

`<replica_set_name>` replica set, , .

, **bindIp** (bind_ip mongo) 0.0.0.0 (0.0.0.0 - ip)

```
  bindIp: 0.0.0.0
```

MongoDB

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

MongoDB

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

replica set

hostname ip-, **/etc/hostname** **/etc/hosts** <IP_address> <hostname>

```
> rs.initiate(
   {
      _id: "replica_set_name",
      version: 1,
      members: [
         { _id: 0, host : "ip_mongo_primary:27017" },
         { _id: 1, host : "ip_mongo_secondary:27017" }
      ]
   }
)
```

Replica Set **Arbiter**:

```
replica_set_name:PRIMARY> rs.add("<ip_server>:27017",true)
{ "ok" : 1 }
```

, shell :

```
replica_set_name:PRIMARY>
```

Replica Set :

```
replica_set_name:PRIMARY> rs.config()
```

Replica Set MongoDB **`rs.status()`**

// Replica Set

Replica Set PRIMARY

Replica Set **Secondary**:

```
replica_set_name:PRIMARY> rs.add("<ip_server>:27017")
{ "ok" : 1 }
```

MongoDB , , (bindIp: `127.0.0.1`), `replication`. .
MongoDB :

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

Replica Set **Arbiter**:

```
replica_set_name:PRIMARY> rs.add("<ip_server>:27017",true)
{ "ok" : 1 }
```

Replica Set (PRIMARY):

```
replica_set_name:PRIMARY> rs.remove("<ip_server>:27017")
{ "ok" : 1 }
```

:

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

Eltex-PCRF

PCRF

PCRF 5701 tcp, 5801 tcp

/etc/eltex-pcrf/hazelcast-cluster-network.xml (5 22 - , 14-15 -)
, :

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

        <!--       -->
        <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">
                <!--  IP-    ( ) -->
                <member>ip_server1</member>
                <member>ip_server2</member>
            </tcp-ip>
            <discovery-strategies>
            </discovery-strategies>
        </join>
        <interfaces enabled="true">
        <!--       -->
            <interface>ip_server1</interface>
        </interfaces>
```

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

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

Eltex-PCRF

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

- 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" : [ ]
}
```


ESR PCRF

PCRF ESR .

SoftWLC

SoftWLC virtual ip . .

:

root@swlc01-server:/# service eltex-<service_name> restart

SoftWLC, , MySQL **localhost 127.0.0.1 <virtual_ip> .**

/etc/eltex-apb/application.conf

```
# maximum number of outgoing messages in queue for each session
sessionMessageQueueSize = 100

# cache config file path
cacheConfigFile = /etc/eltex-apb/ehcache.xml

# path to the file with permitted hosts
hostsFile = /etc/eltex-apb/hosts.json

pingJob {
  # ping job interval
  interval = 60s

  # timeout waiting for subscribe-request after connecting the access point to the server
  subscribeIdleTimeout = 60s
  # timeout during that the session will stay opened without receiving any message
  messageIdleTimeout = 90s
  # interval of ping to be sent to the websocket session
  pingIdleTimeout = 30s
}

# eltex-mercury connection properties
mercury {
  host = localhost
  port = 6565
  poolSize = 50
}
nbi.client.login=admin
nbi.client.password=password
```

- **localhost <virtual_ip>** 24.

/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.ports" : [1813, 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&connectTimeout=5000&socketTimeout=5000&autoReconnect=true&useSSL=false",
```

```
    "user": "javauser",
    "password": "javapassword",
    "max_pool_size": 16
  },

  "mongo.pcrf" : {
    "connection_string": "mongodb://localhost:27017/pcrf?
waitQueueMultiple=500&connectTimeoutMS=10000&socketTimeoutMS=0",
    "db_name": "pcrf"
  },

  "mongo.ott" : {
    "connection_string": "mongodb://localhost:27017/ott?
waitQueueMultiple=500&connectTimeoutMS=10000&socketTimeoutMS=0",
    "db_name": "ott"
  },

  "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&connectTimeout=5000&socketTimeout=5000&autoReconnect=true&useSSL=false",
    "user": "javauser",
    "password": "javapassword",
    "max_pool_size": 16
  },

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

  "sql.auth.service" : {
    "url": "jdbc:mysql://localhost/eltex_auth_service?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&connectTimeout=5000&socketTimeout=5000
&useSSL=false",
    "user": "javauser",
    "password": "javapassword",
    "max_pool_size": 4
  },

  "language" : "en",

  "radius.nbi" : {
    "wdsl.url" : "http://localhost:8080/axis2/services/RadiusNbiService?wsdl",
    "username" : "softwlc_service",
    "password" : "softwlc",
    "connection.timeout.ms" : 30000,
    "request.timeout.ms" : 120000
  },
```

```
  "tariffs.update.interval" : {
    "interval" : 1,
    "unit" : "hours"
  },

  "bras.cron.update.interval": {
    "interval" : 1,
    "unit": "hours"
  },

  "filters.cache.dir" : "/var/lib/eltex-pcrf/filters/",

  "clickhouse": {
    "url": "jdbc:clickhouse://localhost:8123/radius",
    "user_name": "javauser",
    "user_password": "javapassword"
  },

  "accounting.options": {
    "use_clickhouse": false,
    "use_mysql": true,
    "batch_interval_ms": 300000,
    "max_queue_load": 100
  }
}
```

- **mongodb://localhost mongodb://ip_mongo_primary,ip_mongo_secondary** .
- **localhost** **<virtualip>** .
- **127.0.0.1** **<virtualip>** .

/etc/eltex-portal-constructor/application.conf

```
auth-service {
    host = localhost
    port = 21812
    timeout = 10s
    retries = 3
    secret = eltex
    # pap, chap, mschapv2
    protocol = pap
}

login {
    #
    maxAttemptsLogin = 3
    maxAttemptsIP = 5
    #
    blockTime = 5m
}

access {
    //   plaintext-secret   (HMAC256),      FS  PEM- (RSA256)
    secret = "secret"
}

database {
    host = localhost
    port = 3306
    name = ELTEX_PORTAL
    user = javauser
    password = javapassword

    pool {
        # Time to wait for a connection
        connectionTimeout = 10s
        # Time to wait for connection validation
```

```
        validationTimeout = 3s

        min = 1
        max = 10
    }

    cache {
        # Limit of cached simple entries count (for each query type)
        maxEntries = 1000
        # Limit of total cached portal resources size
        maxResourceBytes = 32m
        # Maximum time to retain items in the cache
        expireTime = 30s
    }
}

sso {
    enabled = false
    # Must be in double quotes
    version = "1.0"

    rest {
        scheme = http
        host = localhost
        port = 80
        sso_api_path = /apiman-gateway/b2b_test
    }
    auth {
        scheme = http
        host = localhost
        port = 80
        authentication_path = /auth/realms/b2b/protocol/openid-connect/auth
        logout_path = /auth/realms/b2b/protocol/openid-connect/logout
    }


    params {
        client_id = id
        # URL of epadmin, URL must be in double quotes (!!!)
        redirect_uri = "http://localhost:8080/epadmin/sso"
        client_secret = secret
    }
}

jetty {
    http.port = 9001
    https {
        port = 9444
        keystorePass = 12345
        keystoreFile = /etc/eltex-portal-constructor/localhost.pfx
        keystoreType = PKCS12
        keyAlias = 1
        ciphers = [
            TLS_RSA_WITH_AES_128_CBC_SHA256
            TLS_RSA_WITH_AES_128_CBC_SHA
            TLS_RSA_WITH_AES_256_CBC_SHA256
            TLS_RSA_WITH_AES_256_CBC_SHA
        ]
    }
    multipart {
        maxFileSize = 100MB
        maxRequestSize = 100MB
    }
}

validation {
    public_key = /etc/eltex-doors/keys/public.pem
}

logging {
  host = 100.110.0.212
```

```
  port = 9099
}
```

- **localhost** **<virtualip>** 2, 25, 58, 64, 74,100.

/etc/eltex-portal/application.conf

```
portal {
    defaultRedirectUrl = "http://eltex-co.ru"

    scheduler {
        tariffCheckerPeriod = 1d
        paymentsCleanerPeriod = "0 0 * * * ?"
    }
}

jetty {
    https {
        port = 9443
        keystorePass = 12345
        keystoreFile = /etc/eltex-portal/localhost.pfx
        keystoreType = PKCS12
        keyAlias = 1
        ciphers = [
            TLS_RSA_WITH_AES_128_CBC_SHA256
            TLS_RSA_WITH_AES_128_CBC_SHA
            TLS_RSA_WITH_AES_256_CBC_SHA256
            TLS_RSA_WITH_AES_256_CBC_SHA
        ]
    }
}

database {
    host = localhost
    port = 3306
    name = ELTEX_PORTAL
    user = javauser
    password = javapassword

    pool {
        # Time to wait for a connection
        connectionTimeout = 10s
        # Time to wait for connection validation
        validationTimeout = 3s

        min = 1
        max = 10
    }

    cache {
        # Limit of cached simple entries count (for each query type)
        maxEntries = 1000
        # Limit of total cached portal resources size
        maxResourceBytes = 32m
        # Maximum time to retain items in the cache
        expireTime = 2m
    }
}

// JWT validation. You need a key from Eltex Doors.
// Or you could generate it yourself.
validation {
   public_key = "etc/eltex-doors/keys/public.pem"
```

- **localhost** **<virtualip>** 27.

/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?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=50
00
radius.jdbc.username=javauser
radius.jdbc.password=javapassword
radius.jdbc.maxPoolSize=48
radius.jdbc.inUse=yes

# DB  radius replica(alias=radiusReplicaPool)
#TODO: Change it to replica url
radius.jdbc.replica.driver=org.gjt.mm.mysql.Driver
radius.jdbc.replica.dbUrl=jdbc:mysql://localhost/radius?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=utf8&relaxAutoCommit=true&connectTimeout=50
00
radius.jdbc.replica.username=javauser
radius.jdbc.replica.password=javapassword
radius.jdbc.replica.maxPoolSize=48
radius.jdbc.replica.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=50
00&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=50
00
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=50
00
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=50
00
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=127.0.0.1
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

# wifi-customer-cab request settings
wificab.timeout=90000

# 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

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

# metrics
metric.interval.s=900

################################################################################
##########################DB ELTEX_PORTAL settings##############################
################################################################################
portal.db.driver=com.mysql.jdbc.Driver
portal.db.url=jdbc:mysql://localhost:3306/ELTEX_PORTAL?
max_allowed_packet=32362048&useUnicode=true&characterEncoding=utf8
portal.db.username=javauser
portal.db.password=javapass
```

- **mongodb://localhost mongodb://ip_mongo_primary,ip_mongo_secondary** .
- **localhost <virtualip>** .
- **127.0.0.1 <virtualip>** .

/etc/eltex-ngw/application.conf

```
// Server configuration
server {
    // server port
    port = 8040
    // number of threads in executor that executes handlers and different gateways
    threads = 50
}

http {
    // Timeout of http connection to the end gateway
    connectionTimeout = 30s
    // Number of maximum simultaneous http connections
    maxConnections = 50
    // Time that connection will be kept alive
    keepAliveTimeout = 5s
    // Whether to check SSL certificate
    checkCert = true
    // HTTP User Agent
    userAgent = eltex-ngw

}

sms {
    // Incoming (user to service) sms config
    incoming.config = "smsc.conf"
    // Outgoing (service to user) sms config
    outgoing.config = "smsc.conf"
}

call {
    // Incoming (user to service) call config
    incoming.config = ""
    // Outgoing (service to user) call config
    outgoing.config = ""
}

email {
    // Outgoing (service to user) email config
    outgoing.config = ""

}

database {
    host = localhost
    port = 3306
    name = eltex_ngw
    user = javauser
    password = javapassword

    pool {
        // Time to wait for a connection
        connectionTimeout = 10s
        // Time to wait for connection validation
        validationTimeout = 3s

        min = 1
        max = 10
    }
```

- **localhost virtual_ip** 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"

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

# PCRF
# If you setting pcrf_enabled=0, then you also should enable accounting port listening in "default" server
pcrf_host="127.0.0.1"
pcrf_port=7080
pcrf_enabled=1

# EAP
ca_cert_name="local.pem"
tls_key_password="1234"

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

# Ubiquity vendor detection
ubi_vendor_regex="Apple|Ubiquiti"
vendor_group_enabled=1

# 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

# Proxy SSID (for example to eltex-eap-tls) #139679
proxy_ssid_enabled=0
proxy_ssid_value="EAP_TLS"
proxy_ssid_host="127.0.0.1"
proxy_ssid_port=18122
proxy_ssid_secret="eltex"
```

- **localhost** <virtualip> .
- **127.0.0.1** <virtualip> .

/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://localhost:8080/axis2/services/RadiusNbiService?wsdl</entry>
    <entry key="nbi.serviceLogin.user">softwlc_service</entry>
    <entry key="nbi.serviceLogin.password">softwlc</entry>
    <entry key="nbi.serviceLogin.requestTimeout.sec">120</entry>

    <!--Bonnie or NBI-->
    <entry key="data.service.type">NBI</entry>
    <entry key="bonnie.service.host">localhost</entry>
    <entry key="bonnie.service.port">9070</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>

    <!-- Support link  -->
    <entry key="support.page.enabled">false</entry>
    <entry key="support.page.url">http://eltex-co.ru</entry>

    <!-- DPI link  -->
    <entry key="dpi.page.enabled">false</entry>
    <entry key="dpi.page.url">https://filter.wifi.rt.ru/</entry>

    <!-- SSO Settings -->
    <entry key="sso.enabled">false</entry>
    <entry key="sso.redirectUri">http://localhost:8080/wifi-cab/sso</entry>
    <entry key="sso.clientSecret"></entry>
    <entry key="sso.clientId"></entry>

    <!-- SSO Auth -->
    <entry key="sso.auth.server.protocol">http</entry>
    <entry key="sso.auth.server.address"></entry>
    <entry key="sso.auth.server.port">80</entry>

    <entry key="sso.auth.auth.path">/auth/realms/b2b/protocol/openid-connect/auth</entry>
    <entry key="sso.auth.logout.path">/auth/realms/b2b/protocol/openid-connect/logout</entry>

    <!-- SSO REST -->
    <entry key="sso.rest.server.protocol">http</entry>
    <entry key="sso.rest.server.address"></entry>
    <entry key="sso.rest.server.port">80</entry>
    <entry key="sso.rest.server.timeout.sec">10</entry>
    <entry key="sso.rest.protocol.version">2.0</entry>
    <entry key="sso.rest.username"></entry>
    <entry key="sso.rest.password"></entry>

    <entry key="sso.rest.getToken.path">/apiman-gateway/b2b_test/getToken</entry>
    <entry key="sso.rest.getUserInfo.path">/apiman-gateway/b2b_test/getUserInfo</entry>
    <entry key="sso.rest.addUser.path">/apiman-gateway/b2b_test/addUser</entry>
    <entry key="sso.rest.updateUser.path">/apiman-gateway/b2b_test/updateUser</entry>
    <entry key="sso.rest.delUser.path">/apiman-gateway/b2b_test/delUser</entry>
    <entry key="sso.rest.addUserParam.path">/apiman-gateway/b2b_test/addUserParam</entry>
    <entry key="sso.rest.delUserParam.path">/apiman-gateway/b2b_test/delUserParam</entry>
    <entry key="sso.rest.getUserByName.path">/apiman-gateway/b2b_test/getUserByName</entry>
    <entry key="sso.rest.resetPassword.path">/apiman-gateway/b2b_test/resetPassword</entry>
    <entry key="sso.rest.getUserByParam.path">/apiman-g
```

- **mongodb://localhost mongodb://ip_mongo_primary,ip_mongo_secondary** 4.
- **localhost <virtualip>** .

/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

# memcached server address
#memcached_server_ip_port=127.0.0.1:11211
```

localhost **<virtualip>** 4, 17, 26, 35, 48, 57, 66, 75, 84, 98.

NAS

NAS.

eltex_auth_service , . , .

MySQL :

```
INSERT INTO eltex_auth_service.nas (nasname, shortname, secret, description) VALUES ('<ip_server_1>', 'Server-
1', 'eltex', 'Server-1');
INSERT INTO eltex_auth_service.nas (nasname, shortname, secret, description) VALUES ('<ip_server_2>', 'Server-
2', 'eltex', 'Server-2');
INSERT INTO eltex_auth_service.nas (nasname, shortname, secret, description) VALUES ('<virtual_ip>', 'Virtual
IP', 'eltex', 'Virtual IP');
```

:

- <ip_server_1> - IP- -1
- <ip_server_2> - IP- -2
- <virtual_ip> - IP-

. master-master (-), NAS . , MySQL .

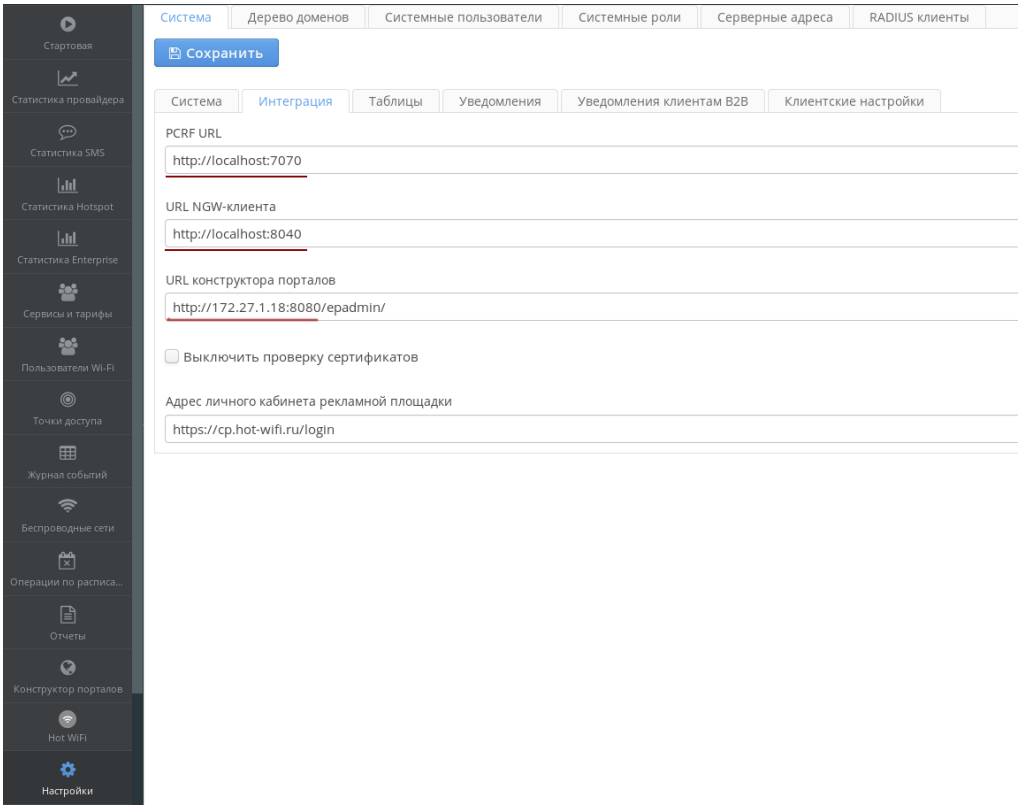
eltex-auth-service.

GUI

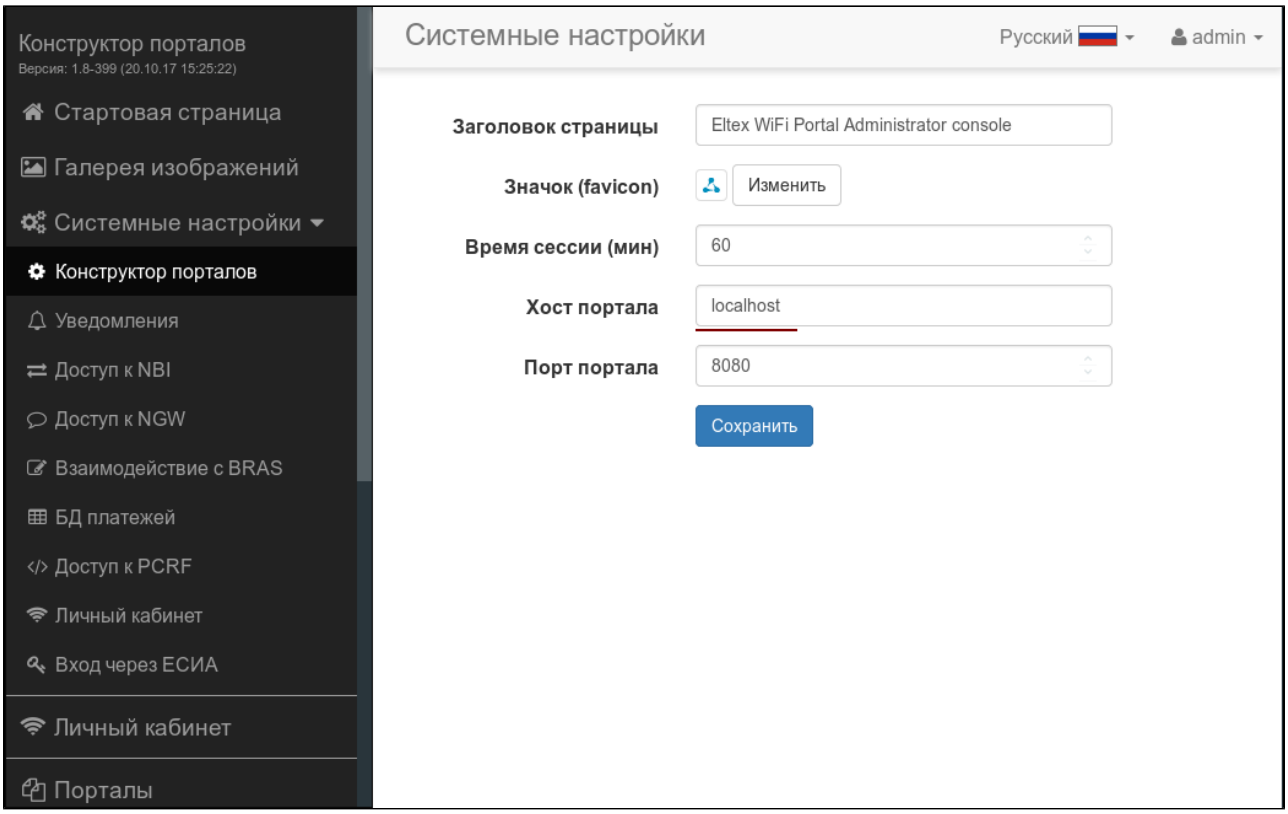
SoftWLC .

Wi-Fi

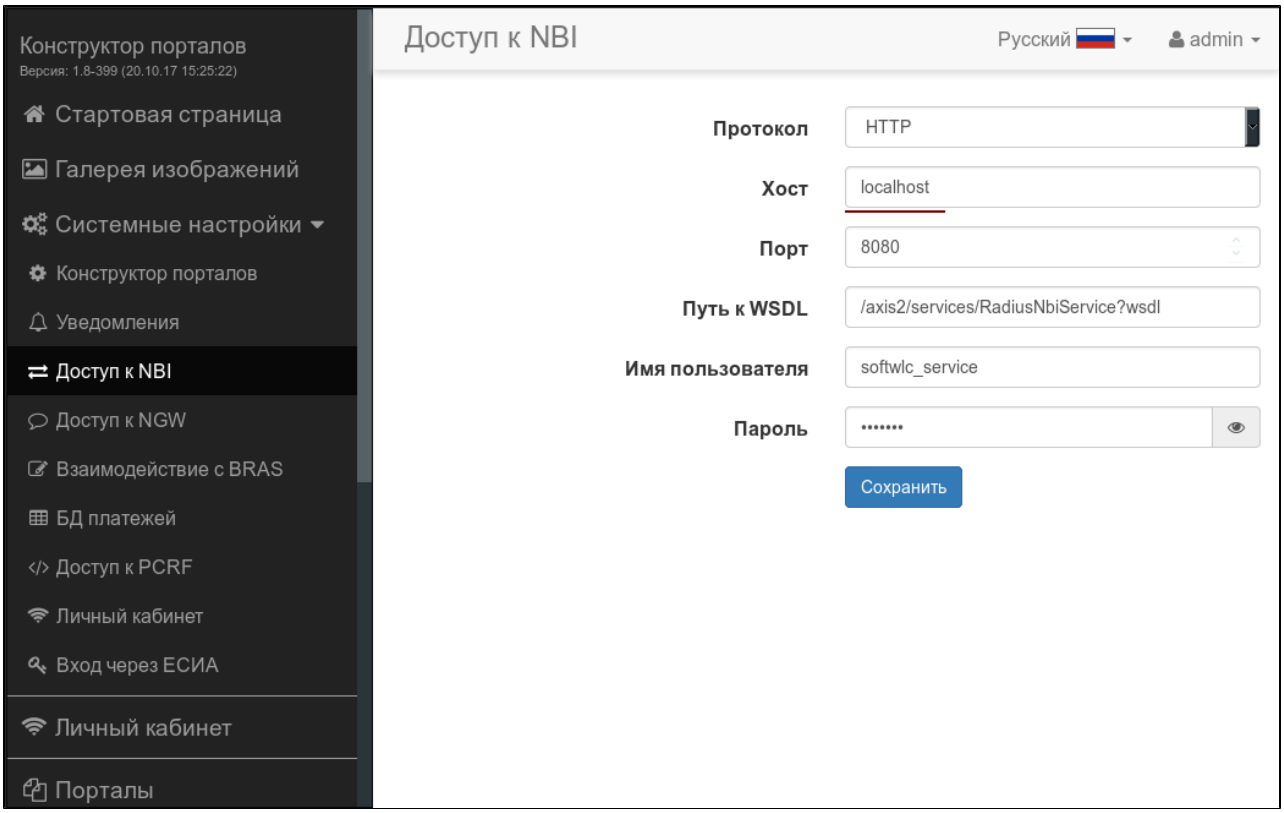
PCRF URL, URL NGW- URL localhost ip-:



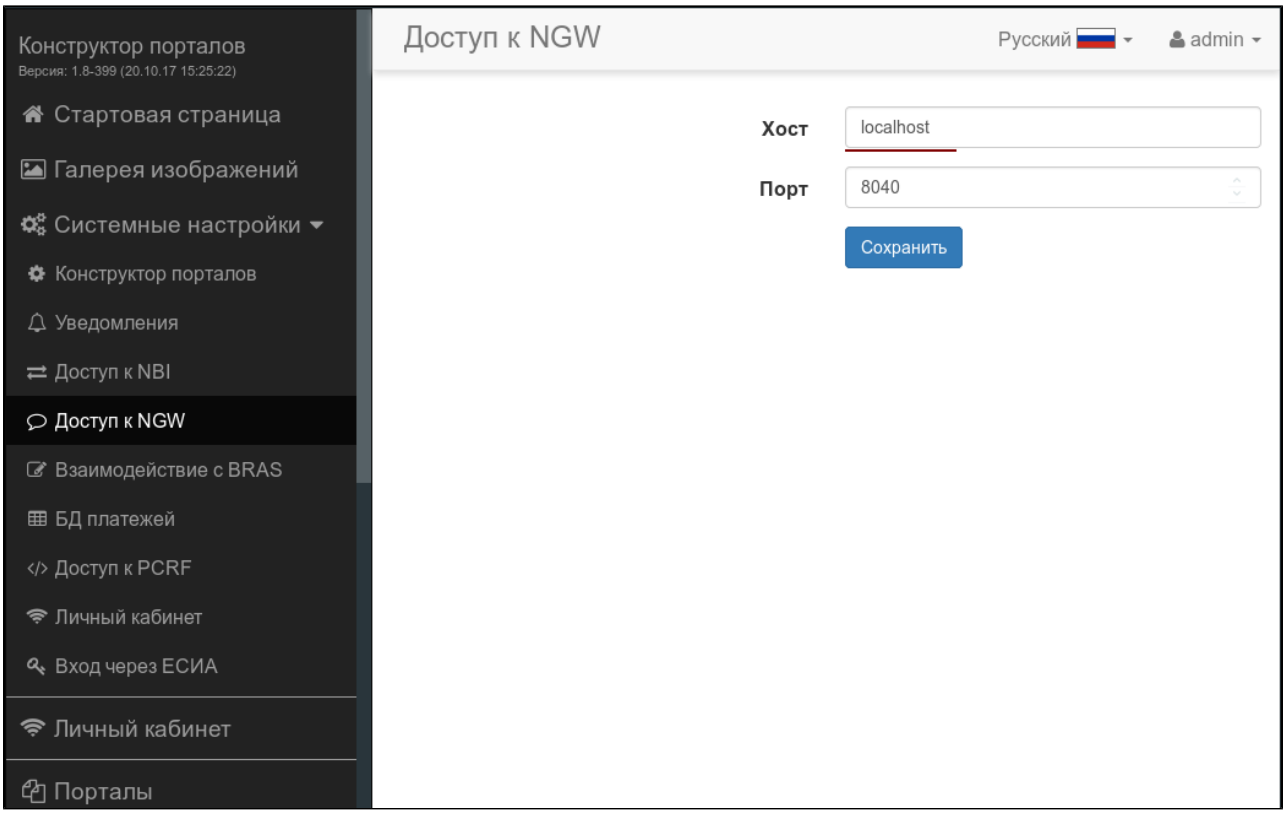
localhost ip- :


Конструктор порталов
Версия: 1.8-399 (20.10.17 15:25:22)
Стартовая страница
Галерея изображений
Системные настройки
Конструктор порталов
Уведомления
Доступ к NBI
Доступ к NGW
Взаимодействие с BRAS
БД платежей
Доступ к PCRF
Личный кабинет
Вход через ЕСИА
Личный кабинет
Порталы
Системные настройки
Русский
admin
Заголовок страницы
Eltex WiFi Portal Administrator console
Значок (favicon)
Изменить
Время сессии (мин)
60
Хост портала
localhost
Порт портала
8080
Сохранить

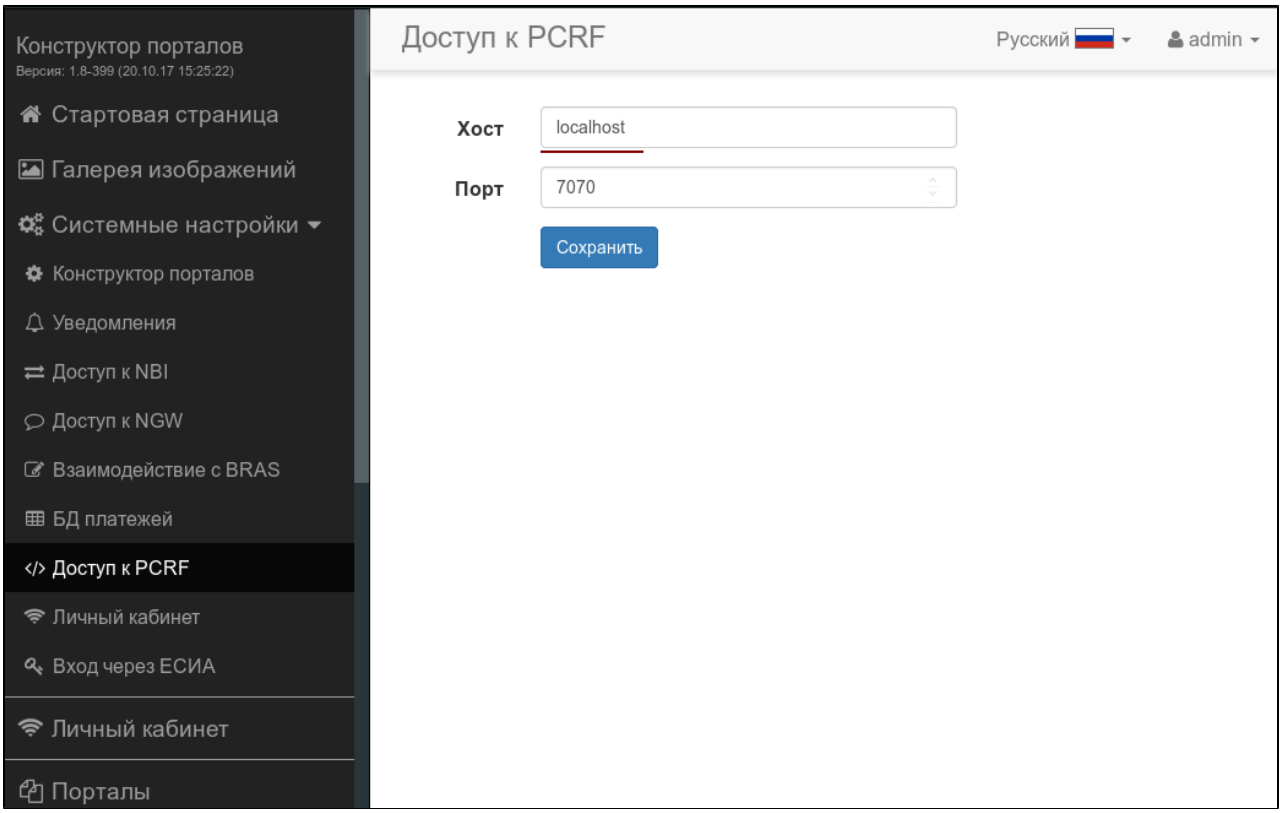
NBI


Конструктор порталов
Версия: 1.8-399 (20.10.17 15:25:22)
Стартовая страница
Галерея изображений
Системные настройки
Конструктор порталов
Уведомления
Доступ к NBI
Доступ к NGW
Взаимодействие с BRAS
БД платежей
Доступ к PCRF
Личный кабинет
Вход через ЕСИА
Личный кабинет
Порталы
Доступ к NBI
Русский
admin
Протокол
HTTP
Хост
localhost
Порт
8080
Путь к WSDL
/axis2/services/RadiusNbiService?wsdl
Имя пользователя
softwlc_service
Пароль
Сохранить

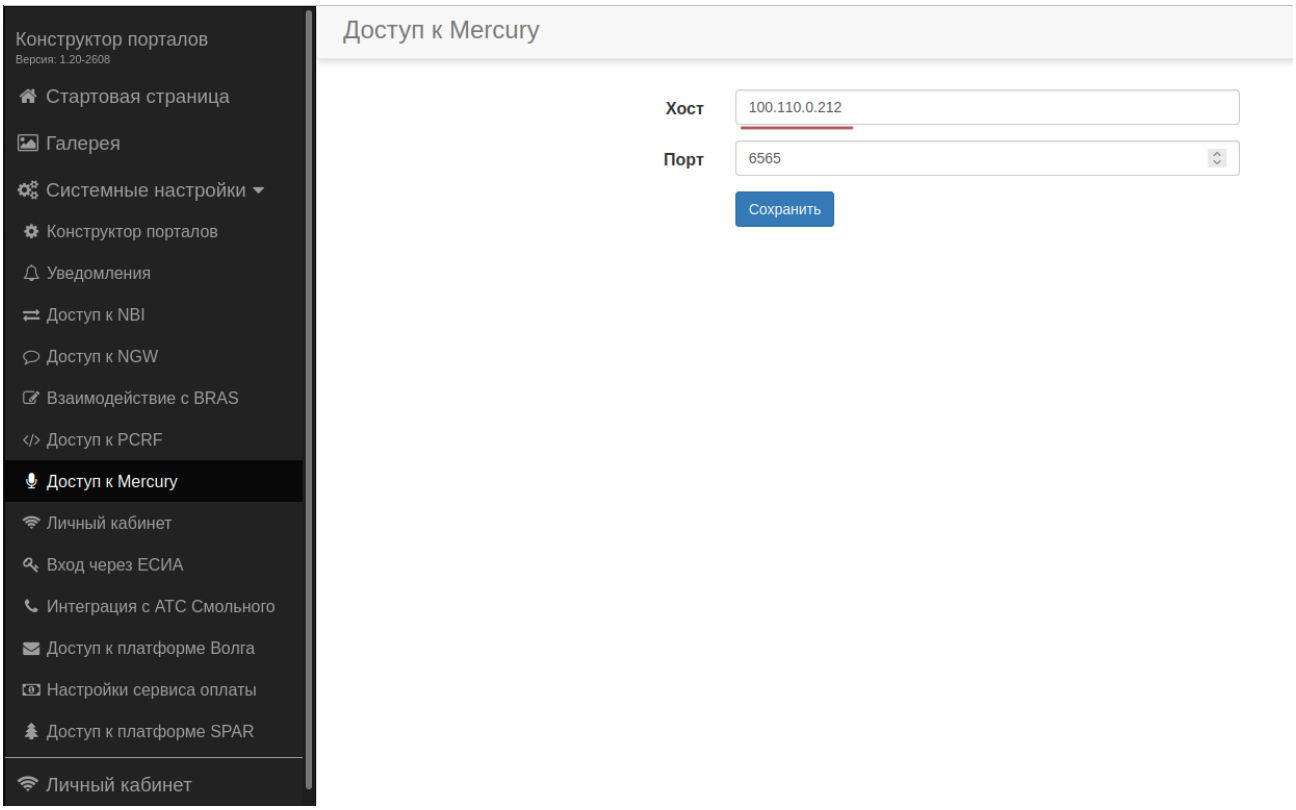
NGW



PCRF



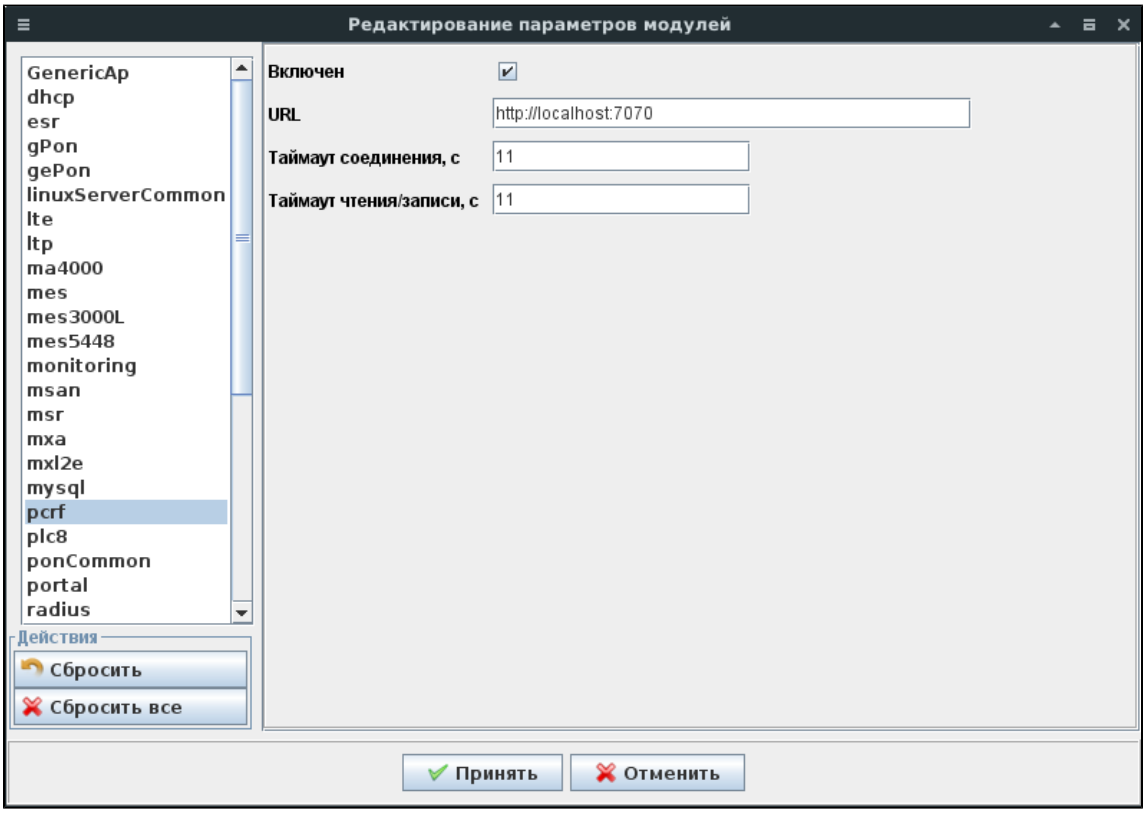
Mercury



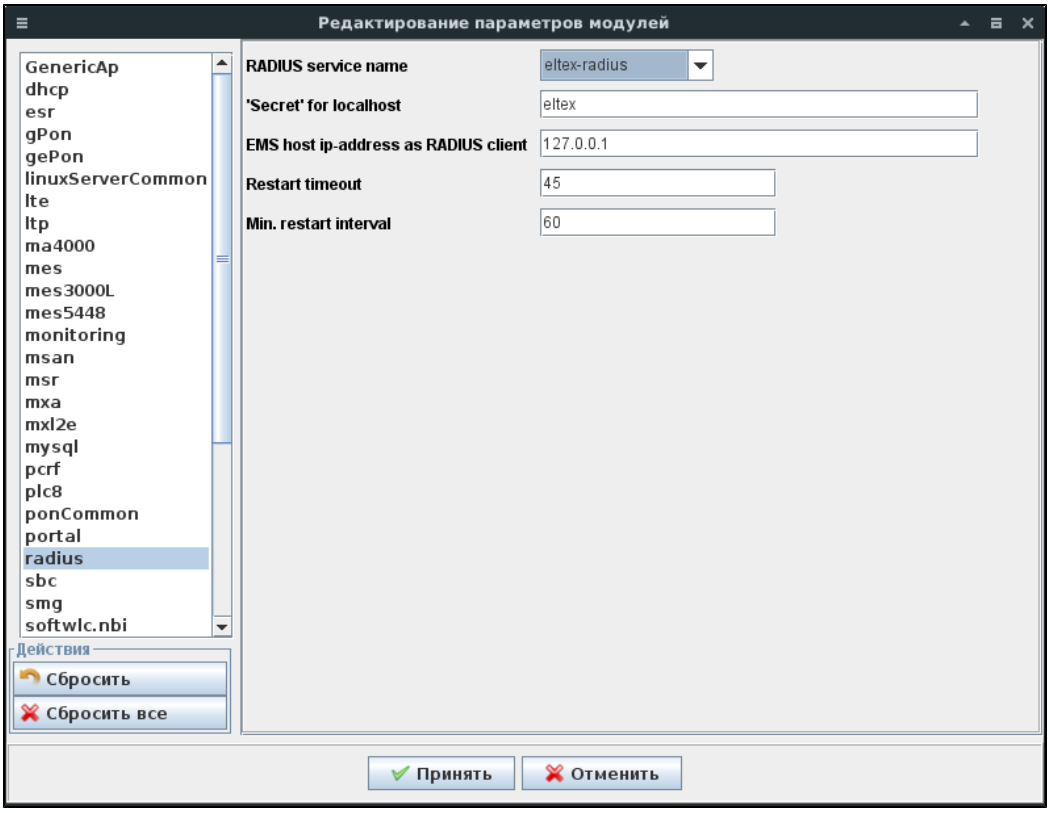
EMS-GUI

EMS localhost (127.0.0.1) ip- :

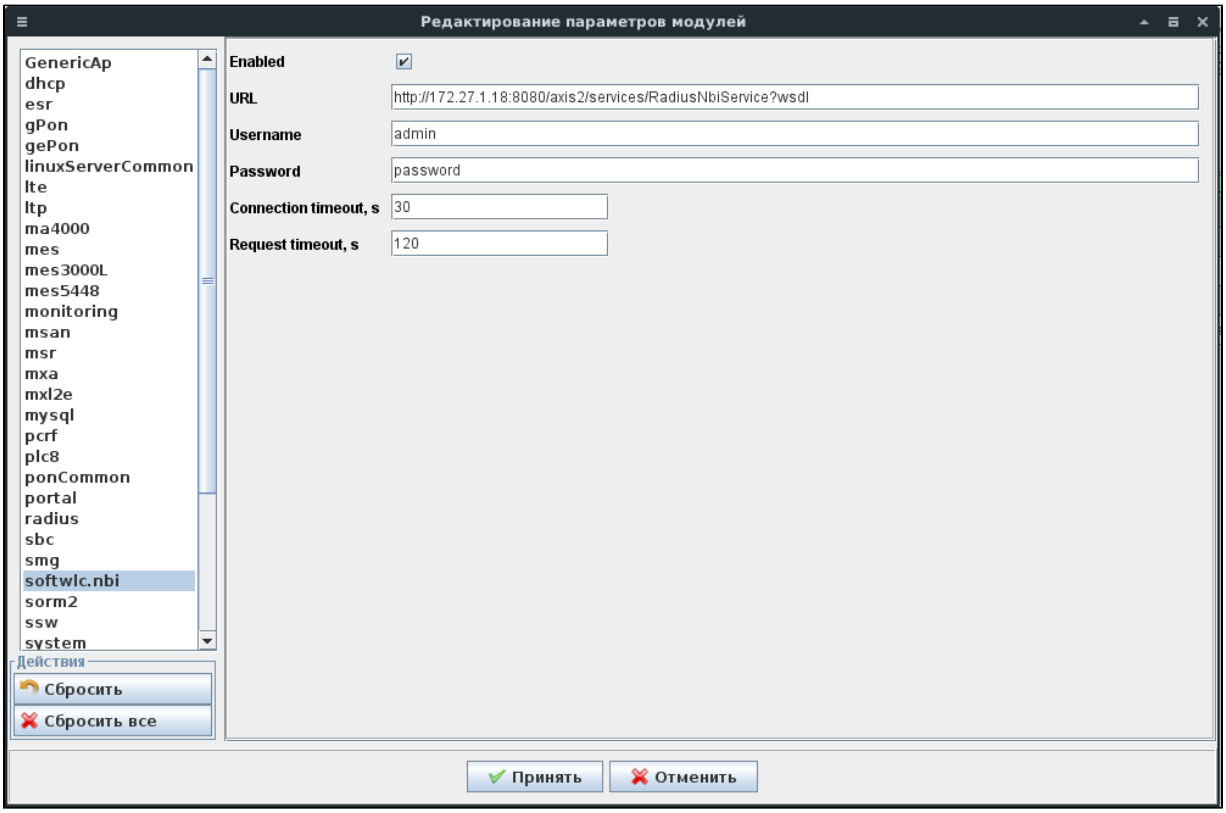
pcrf



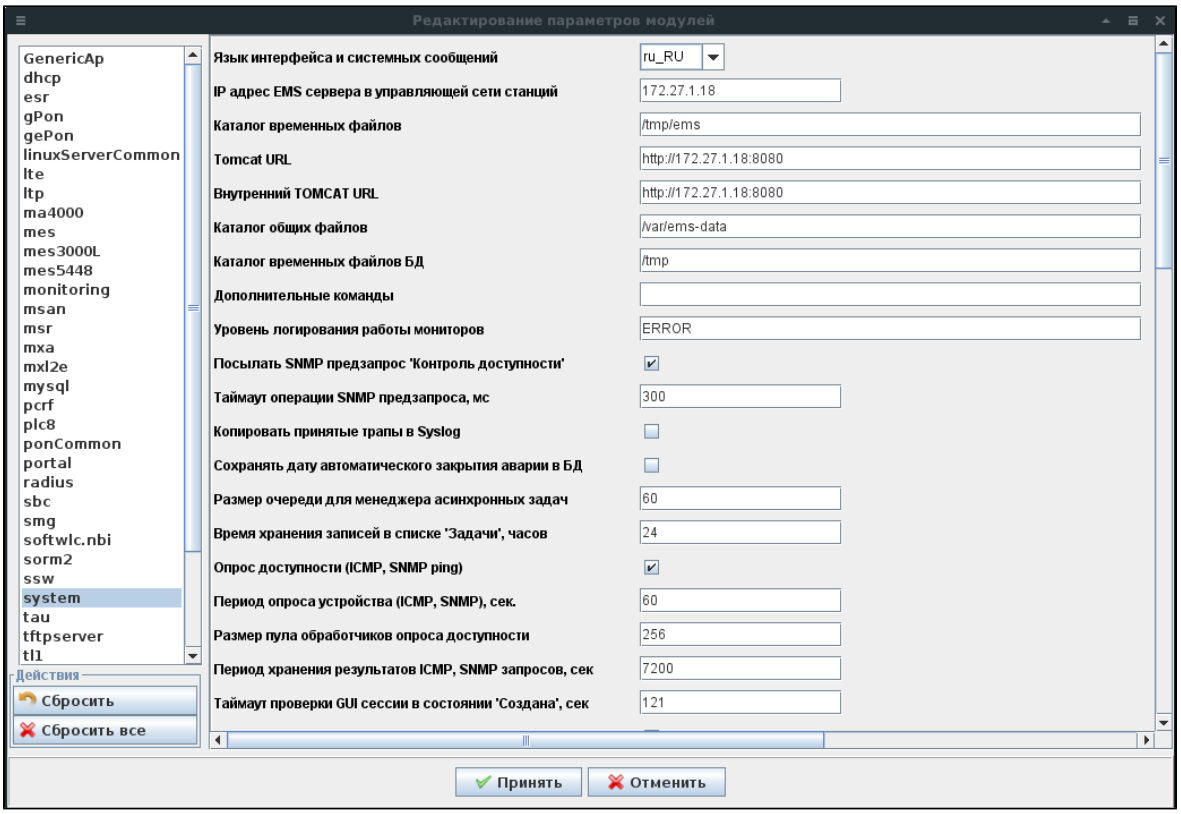
radius



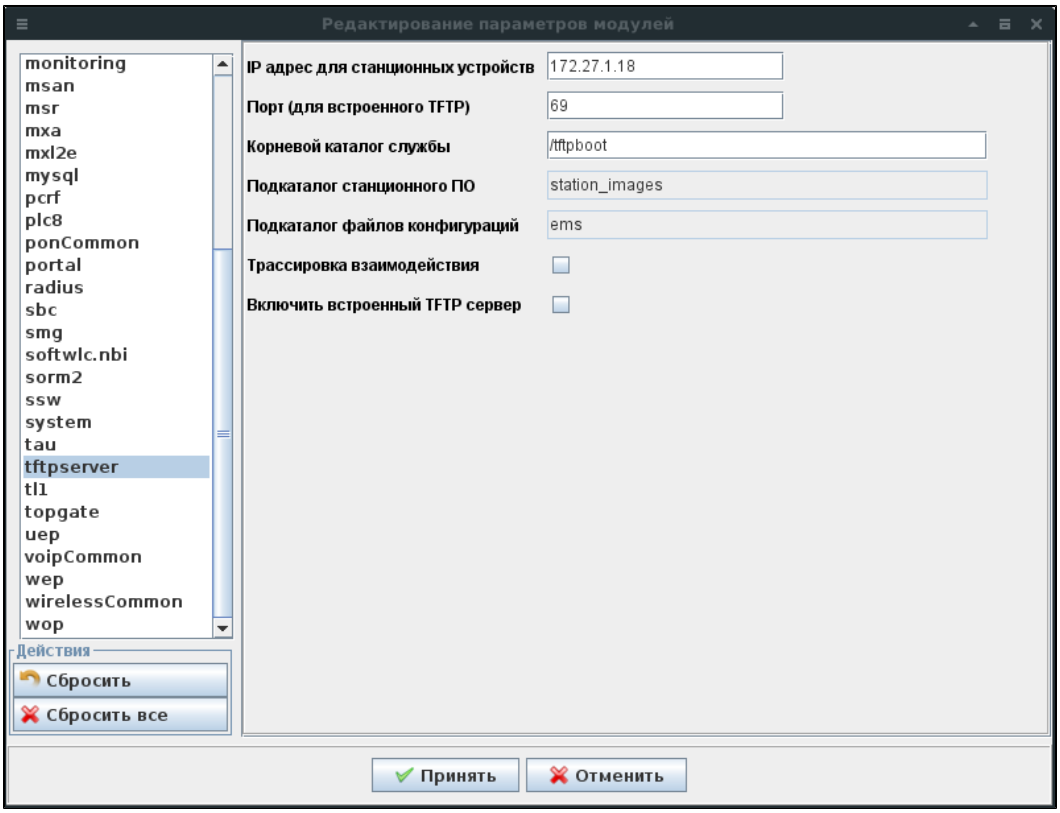
softwlc.nbi



system


Редактирование параметров модулей
GenericAp
dhcp
esr
gPon
gePon
linuxServerCommon
lte
ltp
ma4000
mes
mes3000L
mes5448
monitoring
msan
msr
mxa
mxl2e
mysql
pcrf
plc8
ponCommon
portal
radius
sbc
smg
softwlc.nbi
sorm2
ssw
system
tau
tftpserver
tl1
Действия
Сбросить
Сбросить все
Язык интерфейса и системных сообщений
ru_RU
IP адрес EMS сервера в управляющей сети станций
172.27.1.18
Каталог временных файлов
/tmp/ems
Tomcat URL
http://172.27.1.18:8080
Внутренний TOMCAT URL
http://172.27.1.18:8080
Каталог общих файлов
/var/ems-data
Каталог временных файлов БД
/tmp
Дополнительные команды
Уровень логирования работы мониторов
ERROR
Посылать SNMP предзапрос 'Контроль доступности'
Таймаут операции SNMP предзапроса, мс
300
Копировать принятые трапы в Syslog
Сохранять дату автоматического закрытия аварии в БД
Размер очереди для менеджера асинхронных задач
60
Время хранения записей в списке 'Задачи', часов
24
Опрос доступности (ICMP, SNMP ping)
Период опроса устройства (ICMP, SNMP), сек.
60
Размер пула обработчиков опроса доступности
256
Период хранения результатов ICMP, SNMP запросов, сек
7200
Таймаут проверки GUI сессии в состоянии 'Создана', сек
121
Принять
Отменить

tftpserver


Редактирование параметров модулей
monitoring
msan
msr
mxa
mxl2e
mysql
pcrf
plc8
ponCommon
portal
radius
sbc
smg
softwlc.nbi
sorm2
ssw
system
tau
tftpserver
tl1
topgate
uep
voipCommon
wep
wirelessCommon
wop
Действия
Сбросить
Сбросить все
IP адрес для станционных устройств
172.27.1.18
Порт (для встроенного TFTP)
69
Корневой каталог службы
/tftpboot
Подкаталог станционного ПО
station_images
Подкаталог файлов конфигураций
ems
Трассировка взаимодействия
Включить встроенный TFTP сервер
Принять
Отменить