

v1.24_ SoftWLC

-
- SoftWLC
 -
 -
 - MySQL 'eltex-ems'
 - MySQL RADIUS
 - eltex-wifi-cab, eltex-radius-nbi, eltex-portal-constructor
 - eltex-radius-nbi SoftWLC OSS/BSS
 - eltex-ems
 - eltex-radius
 - eltex-ngw
 - eltex-apb
 - (Authentication Authorization and Accounting) Wi-Fi , BRAS
 - 'Hotspot'
 - Hotspot-
 - WiFi 'Hotspot'
 -
 - B2B,
 - MAC-
 -
 - API Wi-Fi.
 -
 -
 -
 - nginx
 - nginx
 - Nginx tomcat8
- SoftWLC
- MySQL
- Mongo
 - // Replica Set
- Eltex-PCRF
-
-
-

 .

SoftWLC eltex, .

:

```
vagrant@ubuntu:~$ dpkg -l | grep eltex-
```

- (Stable_versions) .

- , , .

, , .

snapshot .

backUP .

pv .

login/password .

```
vagrant@ubuntu:~$ sudo apt install pv # pv
vagrant@ubuntu:~$ sudo chmod +x backup_mysql.sh # mysql
vagrant@ubuntu:~$ sudo ./backup_mysql.sh root root # mysql root, root
vagrant@ubuntu:~$ sudo chmod +x mongo_update.sh. # mongo
vagrant@ubuntu:~$ sudo ./mongo_update.sh --backup # mongo
```

[mongo_update.sh](#)

[backup_mysql.sh](#)

BackUP SoftWLC. , backup SoftWLC 1 SoftWLC 2.

backup

```
vagrant@ubuntu:~$ sudo cp /etc/apt/sources.list.d/eltex.list /etc/apt/sources.list.d/eltex.list.  
backup
```

SoftWLC



MySQL:

```
vagrant@ubuntu:~$ mysql -uroot -proot  
mysql> stop slave;  
Query OK, 0 rows affected (0.01 sec)
```

· · ·

SoftWLC eltex, .

('tee'):

```
vagrant@ubuntu:~$ dpkg -l | grep eltex- | tee eltex-packets.txt
```

- ([Stable_versions](#)) .



```
SoftWLC 1.18 ,            1.19.  
notification-gw (   , e-mail   .. ).  
1.19            .  
1.19            1.19:
```

```
vagrant@ubuntu:~$ echo "deb [arch=amd64] http://archive.eltex-co.ru/wireless softwlc-1.19-xenial main" |  
sudo tee /etc/apt/sources.list.d/eltex.list
```

```
vagrant@ubuntu:~$ sudo echo "deb [arch=amd64] http://archive.eltex-co.ru/wireless softwlc-1.24-xenial main" >>  
/etc/apt/sources.list.d/eltex.list  
vagrant@ubuntu:~$ sudo echo "deb [arch=amd64] http://archive.eltex-co.ru/wireless radius-1.24-bionic mai" >>  
/etc/apt/sources.list.d/eltex.list
```

```
vagrant@ubuntu:~$ sudo apt-get update
```

MySQL 'eltex-ems'

```

sudo apt-get install eltex-ems-db
Enter your MySQL administrator login (default is 'root'): root
Enter your MySQL administrator password:
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
Backup databases? (Y/n) n

```



: Got error: 1102: Incorrect database name '#mysql50#example.database' when selecting the database
:

```

#ls -la /var/lib/mysql | grep ^d | egrep '\.'
#mv /var/lib/mysql/.directory /root/

```

MySQL RADIUS

```

apt-get install eltex-radius-db
Enter your MySQL administrator login (default is 'root'): root
Enter your MySQL administrator password:
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
Backup databases? (Y/n) n

```

eltex-wifi-cab, eltex-radius-nbi, eltex-portal-constructor

```

apt-get install eltex-auth-service-db
Enter your MySQL administrator login (default is 'root'): root
Enter your MySQL administrator password:

```



, .
/, .

Configuration file '/etc/eltex-radius/local.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
 What would you like to do about it ? Your options are:
 Y or I : install the package maintainer's version
 N or O : keep your currently-installed version
 D : show the differences between the versions
 Z : start a shell to examine the situation
 The default action is to keep your current version.
 *** local.conf (Y/I/N/O/D/Z) [default=N] ? D (n)

eltex-radius-nbi SoftWLC OSS/BSS

```

apt-get install eltex-radius-nbi
Do you want to generate server certificate? [y/N]: N

```

eltex-ems

```

apt-get install eltex-ems

```

eltex-radius

```
apt-get install eltex-radius
```

eltex-ngw

```
apt-get install eltex-ngw
```

eltex-apb

```
apt-get install eltex-apb
```

(Authentication Authorization and Accounting) Wi-Fi , BRAS

```
apt-get install eltex-pcrf
```

'Hotspot'

```
apt-get install eltex-logging-service
```

Hotspot-

```
apt-get install eltex-mercury
```

WiFi 'Hotspot'

```
apt-get install eltex-portal
```

```
apt-get install eltex-portal-constructor
```

B2B,

```
apt-get install eltex-wifi-cab
```

MAC-

```
apt-get install eltex-oui-list
```

```
apt-get install eltex-disconnect-service
```

API Wi-Fi.

```
apt-get install eltex-johnny
```

```
apt-get install eltex-bruce
```

```
apt-get install eltex-bruce
```

```
apt-get install eltex-jobs
```

nginx

```
wget http://archive.eltex-co.ru/wireless/nginx/conf/softwlc_<__softwlc>_nginx.conf -O softwlc_<__softwlc>_nginx.conf
```

nginx

```
cp softwlc_<__softwlc>_nginx.conf /etc/nginx/conf.d/softwlc.conf
```

Nginx tomcat8

```
sudo systemctl restart nginx  
sudo systemctl restart tomcat8
```

SoftWLC

SoftWLC virtual ip.



:

```
tester@ubuntu:~$ sudo service eltex-<service_name> restart
```

/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?
replicaSet=<YourClusterName>&waitQueueMultiple=500&connectTimeoutMS=10000&socketTimeoutMS=0&readPreference=secondaryPreferred",
    "db_name": "pcrf"
  },

  "mongo.ott" : {
    "connection_string": "mongodb://localhost:27017/ott?
replicaSet=<YourClusterName>&waitQueueMultiple=500&connectTimeoutMS=10000&socketTimeoutMS=0&readPreference=secondaryPreferred",
    "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" : {
    "wds1.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 replicaSet, /etc/mongod.conf.`

```

mongodb://192.168.10.3:27017,192.168.10.4:27017/pcrf?
replicaSet=Cluster&waitQueueMultiple=500&connectTimeoutMS=10000&socketTimeoutMS=0&readPreference=secondaryPrefer
red

mongodb://192.168.10.3:27017,192.168.10.4:27017/ott?
replicaSet=Cluster&waitQueueMultiple=500&connectTimeoutMS=10000&socketTimeoutMS=0&readPreference=secondaryPrefer
red

```

- `localhost <virtualip> , 14-`
- `127.0.0.1 <virtualip> , 14-`

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

```

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 = localhost

```

```
port = 9099
}
```

- **localhost** <virtualip> 15, 48, 54, 64, 95.

/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?replicaSet=<YourClusterName>

# wifi-customer-cab mongodb connector
wificab.mongodb.enabled=true
wificab.mongodb.uri=mongodb://localhost:27017/wifi-customer-cab?replicaSet=<YourClusterName>

# Eltex.SORM2.replicator MongoDB 'sorm2' connect
sorm2.mongodb.enabled=false
sorm2.mongodb.uri=mongodb://localhost:27017/sorm2?replicaSet=<YourClusterName>

# 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?replicaSet=<YourClusterName>

# 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 replicaSet, /etc/mongod.conf.`

```
pcrf.mongodb.uri=mongodb://192.168.10.3:27017,192.168.10.4:27017/pcrf?replicaSet=Cluster
wificab.mongodb.uri=mongodb://192.168.10.3:27017,192.168.10.4:27017/wifi-customer-cab?replicaSet=Cluster
sorm2.mongodb.uri=mongodb://192.168.10.3:27017,192.168.10.4:27017/sorm2?replicaSet=Cluster
ott.mongodb.uri=mongodb://192.168.10.3:27017,192.168.10.4:27017/ott?replicaSet=Cluster
```

- `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 = "smssc.conf"
    // Outgoing (service to user) sms config
    outgoing.config = "smssc.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 44.**

/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"

# Ubiquiti 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?replicaSet=<YourClusterName></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">password</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>

  <!--Bruce-->
  <entry key="bruce.service.host">localhost</entry>
  <entry key="bruce.service.port">8008</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.example.org/</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-gateway/b2b_test/getUserByParam</entry>
  <entry key="sso.rest.getUserByEmail.path">/apiman-gateway/b2b_test/getUserByEmail</entry>
</properties>
```

- `mongodb://localhost mongodb://ip_mongo_primary,ip_mongo_secondary 4 replicaSet, /etc/mongod.conf.`


```
<entry key="mongoaddress">mongodb://192.168.10.3:27017,192.168.10.4:27017/wifi-customer-cab?
replicaSet=Cluster</entry>
```

- 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.



2- .

MySQL

:

mysql

, :(2-)

```
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.002455
      Read_Master_Log_Pos: 89717415
      Relay_Log_File: mysqld-relay-bin.000004
      Relay_Log_Pos: 89717561
      Relay_Master_Log_File: mysql-bin.002455
      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:
```

- .

Mongo

:

mongo

Replica Set :

```
replica:PRIMARY> rs.status()
{
  "set" : "replica",
  "date" : ISODate("2023-01-10T09:30:20.890Z"),
  "myState" : 1,
  "term" : NumberLong(24),
  "syncingTo" : "",
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1673343020, 1),
      "t" : NumberLong(24)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1673343020, 1),
      "t" : NumberLong(24)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1673343020, 1),
      "t" : NumberLong(24)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1673343020, 1),
      "t" : NumberLong(24)
    }
  },
  "lastStableCheckpointTimestamp" : Timestamp(1673342958, 1),
  "electionCandidateMetrics" : {
    "lastElectionReason" : "electionTimeout",
    "lastElectionDate" : ISODate("2023-01-10T05:31:08.612Z"),
    "electionTerm" : NumberLong(24),
    "lastCommittedOpTimeAtElection" : {
      "ts" : Timestamp(1673328656, 1),
```

```

        "t" : NumberLong(22)
    },
    "lastSeenOpTimeAtElection" : {
        "ts" : Timestamp(1673328656, 1),
        "t" : NumberLong(22)
    },
    "numVotesNeeded" : 2,
    "priorityAtElection" : 2,
    "electionTimeoutMillis" : NumberLong(10000),
    "numCatchUpOps" : NumberLong(0),
    "newTermStartDate" : ISODate("2023-01-10T05:31:08.617Z"),
    "wMajorityWriteAvailabilityDate" : ISODate("2023-01-10T05:31:09.488Z")
},
"electionParticipantMetrics" : {
    "votedForCandidate" : true,
    "electionTerm" : NumberLong(22),
    "lastVoteDate" : ISODate("2023-01-10T04:15:15.939Z"),
    "electionCandidateMemberId" : 1,
    "voteReason" : "",
    "lastAppliedOpTimeAtElection" : {
        "ts" : Timestamp(1673324064, 1),
        "t" : NumberLong(17)
    },
    "maxAppliedOpTimeInSet" : {
        "ts" : Timestamp(1673324064, 1),
        "t" : NumberLong(17)
    },
    "priorityAtElection" : 2
},
"members" : [
    {
        "_id" : 0,
        "name" : "100.110.1.7:27017",
        "health" : 1,
        "state" : 1,
        "stateStr" : "PRIMARY",
        "uptime" : 20400,
        "optime" : {
            "ts" : Timestamp(1673343020, 1),
            "t" : NumberLong(24)
        },
        "optimeDate" : ISODate("2023-01-10T09:30:20Z"),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "electionTime" : Timestamp(1673328668, 1),
        "electionDate" : ISODate("2023-01-10T05:31:08Z"),
        "configVersion" : 75844,
        "self" : true,
        "lastHeartbeatMessage" : ""
    },
    {
        "_id" : 1,
        "name" : "100.110.0.246:27017",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 14361,
        "optime" : {
            "ts" : Timestamp(1673343018, 1),
            "t" : NumberLong(24)
        },
        "optimeDurable" : {
            "ts" : Timestamp(1673343018, 1),
            "t" : NumberLong(24)
        },
        "optimeDate" : ISODate("2023-01-10T09:30:18Z"),
        "optimeDurableDate" : ISODate("2023-01-10T09:30:18Z"),
        "lastHeartbeat" : ISODate("2023-01-10T09:30:20.114Z"),
        "lastHeartbeatRecv" : ISODate("2023-01-10T09:30:19.491Z"),
    }
]

```

```

        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "100.110.1.7:27017",
        "syncSourceHost" : "100.110.1.7:27017",
        "syncSourceId" : 0,
        "infoMessage" : "",
        "configVersion" : 75844
    },
    {
        "_id" : 2,
        "name" : "100.110.0.206:27017",
        "health" : 1,
        "state" : 7,
        "stateStr" : "ARBITER",
        "uptime" : 12138,
        "lastHeartbeat" : ISODate("2023-01-10T09:30:20.704Z"),
        "lastHeartbeatRecv" : ISODate("2023-01-10T09:30:19.957Z"),
        "pingMs" : NumberLong(1),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 75844
    }
],
"ok" : 1,
"operationTime" : Timestamp(1673343020, 1),
"$clusterTime" : {
    "clusterTime" : Timestamp(1673343020, 1),
    "signature" : {
        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
        "keyId" : NumberLong(0)
    }
}
}

```

3 .

, .

// 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 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" : [ ]
}
```

, SoftWLC , .

, , backup Mongo MySQL.

downgrade .

1. tester@ubuntu:~\$ sudo mv /etc/apt/sources.list.d/eltex.list.backup /etc/apt/sources.list.d/eltex.list
2. tester@ubuntu:~\$ sudo apt update
3. (eltex-auth-service-db, eltex-ems-db, eltex-radius-db ..)

apt install eltex-ems-db
 apt install eltex-auth-service-db
 apt install eltex-radius-db

1. ()

apt install eltex-ems
 apt install eltex-auth-service
 apt install eltex-radius

backup Mysql Mongo .

mongod.conf mongo_update.sh

```
sudo ./mongo_update.sh --restore-config
sudo ./mongo_update.sh --restore
sudo chmod +x restore_mysql.sh
sudo ./restore_mysql.sh root root
```

[restore_mysql.sh](#)

[mongo_update.sh](#)