v1.20_PCRF

-
-
  -
  - /
-
  - /etc/default/eltex-pcrf
  - /etc/eltex-pcrf/eltex-pcrf.json
  - /etc/eltex-pcrf/hazelcast-cluster-network.xml
  - /etc/eltex-pcrf/log4j.xml

Comala Metadata License Details

Invalid commercial evaluation license with a expired error. Please click here to purchase a commercial license.

:

- , Wi-Fi , BRAS (,).
- .
- Wi-Fi .
- Wi-Fi , WPA-enterprise BRAS.

'http://<ip address pcrf>:7070/pcrf'.

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

/

	`service eltex-pcrf status`	`* eltex-pcrf process is running` `* eltex-pcrf process is not running`

	`service eltex-pcrf start`	`* Starting eltex-pcrf`
		
		`* eltex-pcrf is already running`

	`service eltex-pcrf stop`	`* Stopping eltex-pcr`
		
		`,..`
		`* eltex-pcrf is not running`

	`service eltex-pcrf restart`	`* Stopping eltex-pcrf`
		`* Starting eltex-pcrf`
		
		`* eltex-pcrf is not running`
		`* Starting eltex-pcrf`

/etc/default/eltex-pcrf

. :

/etc/default/eltex-pcrf

```
# Eltex.PCRF Server daemon parameters
NAME="eltex-pcrf"

# Location of java binary
JAVA=/usr/bin/java

# Initial size of Java heap
JAVA_INIT_HEAP=256m
# Maximum size of Java heap
JAVA_MAX_HEAP=512m

# Options for Java Garbage Collector
GC_OPTS="-XX:+UseG1GC \
-XX:+PrintGCDateStamps \
-XX:+PrintGCDetails \
-XX:+UseGCLogFileRotation \
-XX:NumberOfGCLogFiles=7 \
-XX:GCLogFileSize=5M \
-XX:+UseStringDeduplication \
-XX:+PrintGCTimeStamps \
-XX:+PrintTenuringDistribution \
-Xloggc:/var/log/eltex-pcrf/gc.log"

# To monitor via JMX - jconsole to host:port
#JMX_OPTS="-Dcom.sun.management.jmxremote \
#-Dcom.sun.management.jmxremote.port=8085 \
#-Dcom.sun.management.jmxremote.authenticate=false \
#-Dcom.sun.management.jmxremote.ssl=false \
#-Djava.rmi.server.hostname=127.0.0.1"

HEAP_DUMP_OPTS="-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/var/log/eltex-pcrf"
VERTX_CACHE_DIR_OPT="-Dvertx.cacheDirBase=/var/lib/$NAME"

JVM_OPTS="$JMX_OPTS $GC_OPTS $HEAP_DUMP_OPTS -Xms$JAVA_INIT_HEAP -Xmx$JAVA_MAX_HEAP $VERTX_CACHE_DIR_OPT"
JAR="/usr/lib/$NAME/$NAME.jar"

# Send notification
#NOTIFICATION_ADDR="admin@mail.loc
```

JAVA=/usr/bin/java	Java-.
JAVA_INIT_HEAP	, . JAVA_MAX_HEAP.
JAVA_MAX_HEAP	, .
GC_OPTS="-XX:+UseG1GC \ . . .	.
# To monitor via JMX - jconsole to host:port . . .	JMX.
HEAP_DUMP_OPTS	.
VERTX_CACHE_DIR_OPT	.
JVM_OPTS	jvm.
NOTIFICATION_ADDR	.

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

.

/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,
    "coa.secret" : "testing123",
    "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",
    "actualization_period_ms": 15000
  },

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

  "generic.ap.registrar": {
    "max_aps_in_queue": 30,
    "ap_register_interval_ms": 600000,
    "added_ap_cache_ttl_ms": 600000,
    "host": "localhost",
    "port": 8080,
    "worker_pool_size": 8
  },
```

```
  "kafka": {
    "mcd.enabled": false,
    "circuit.breaker": {
      "timeout.ms": 30000,
      "reset.ms": 60000,
      "max.failures": 3
    },
    "producer": {
      "bootstrap.servers": "localhost:9092",
      "linger.ms": "1000",
      "topic": "mcd",
      "max.block.ms": "30000",
      "request.timeout.ms": "14000",
      "delivery.timeout.ms": "30000",
      "acks": "1",
      "retries": "1"
    }
  }
}
```

auth.address	, .
auth.port	, .
auth.mac.open.timeout.s	"", , .
auth.mac.welcome.service	, .
acct.address	, .
acct.port	, .
Wi-Fi BRAS	
aaa.host	Eltex.RADIUS.
aaa.secret	RADIUS key.
aaa.auth.port	, .
aaa.acct.port	, .
aaa.rest.port	Eltex.RADIUS.
aaa.timeout	.
aaa.attempts	.
web.monitoring.port	.

cluster.enable	.
cluster.eventBusPort	.
MySQL ('radius', 'eltex_ems', 'wireless', 'auth.service')	
url	Mysql.
user	.
password	.
max_pool_size	, .
Mongo('pcrf', 'ott')	
connection_string	URI Mongo.
db_name	, .
Mongo	
session.check.period.s	. , .
unauth.store.time.s	, BRAS, .
interval.number.expired	, "" .
min.interval.s	, .
default.interval.s	, .
CoA	
coa.timeout	.
coa.attempts	.
coa.secret	.
remote.coa.port	, .
log.clean.period	.
log.store.period period unit	. . (- "D").
worker.pool.size	CoA
language	, , "en". "ru", .
NBI	
wdsl.url	NBI.

`username`	.
`password`	.
`connection.timeout.ms`	.
`request.timeout.ms`	.
`interval`	.
`unit`	("hours").
URL	
filters.cache.dir	.
Yandex ClickHouse	
url	Yandex ClickHouse.
user_name	.
user_password	.
actualization_period_ms	.
use_clickhouse	Yandex ClickHouse.
use_mysql	MySQL.
batch_interval_ms	.
max_queue_load	.
generic AP, option 82	
max_aps_in_queue	generic AP , EMS.
ap_register_interval_ms	(), generic AP EMS, .
added_ap_cache_ttl_ms	(), generic AP EMS.
host	IP-, EMS-NBI.
port	, EMS-NBI.
worker_pool_size	, GenericAP.
Apache Kafka	
mcd.enabled	.
circuit.breaker	
timeout.ms	,
max.failures	. , **max.failures**, **reset.ms**.
reset.ms	, . . , . , .
producer	
bootstrap.servers	.

linger.ms	.
topic	.
max.block.ms	.
request.timeout.ms	.
delivery.timeout.ms	.
acks	.
retries	delivery.timeout.ms.

/etc/eltex-pcrf/hazelcast-cluster-network.xml

Hazelcast, PCRF.

/etc/eltex-pcrf/hazelcast-cluster-network.xml

```
<hazelcast xsi:schemaLocation="http://www.hazelcast.com/schema/config http://www.hazelcast.com/schema/config
/hazelcast-config-3.6.xsd"
           xmlns="http://www.hazelcast.com/schema/config"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <!-- You can separate your clusters in a simple way by specifying group names. -->
    <group>
        <name>dev</name>
    </group>

    <network>
        <!-- Write here public address of the node -->
        <public-address>192.168.0.1</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">
                <!-- Write here IP of all members of the cluster (including this) -->
                <member>192.168.0.1</member>
                <member>192.168.0.2</member>
            </tcp-ip>
            <discovery-strategies>
            </discovery-strategies>
        </join>
        <interfaces enabled="true">
            <!-- Write here IP of the interface to use for cluster -->
            <interface>192.168.0.1</interface>
        </interfaces>
        <ssl enabled="false"/>
        <socket-interceptor enabled="false"/>
        <symmetric-encryption enabled="false">
            <algorithm>PBEWithMD5AndDES</algorithm>
            <!-- salt value to use when generating the secret key -->
            <salt>thesalt</salt>
            <!-- pass phrase to use when generating the secret key -->
            <password>thepass</password>
            <!-- iteration count to use when generating the secret key -->
            <iteration-count>19</iteration-count>
        </symmetric-encryption>
    </network>
</hazelcast>
```


<name>dev</name>	. . , .
<public-address>192.168.0.1</public-address>	.
<member>192.168.0.1</member> <member>192.168.0.2</member>	, . , .
<interface>192.168.0.1</interface>	(public-address).

/etc/eltex-pcrf/log4j.xml

.

/etc/eltex-pcrf/log4j.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<Configuration packages="biz.paluch.logging.gelf.log4j2">
    <Properties>
        <Property name="maxSize" value="5MB"/>
        <Property name="maxCount" value="7"/>
        <Property name="logDir" value="/var/log/eltex-pcrf"/>
        <Property name="defaultPattern" value="%d{ISO8601} [%t] %-5p %logger{1} %C{1}.%M(line:%L). %m%n"/>
        <Property name="gelfHost" value="udp:lab3-test.eltex.loc"/>
        <Property name="gelfPort" value="12201"/>
        <Property name="gelfLevel" value="OFF"/>
        <Property name="filenamePrefix" value="eltex-pcrf-"/>
    </Properties>

    <Appenders>
        <Console name="STDOUT" target="SYSTEM_OUT">
            <PatternLayout>
                <Pattern>${defaultPattern}</Pattern>
            </PatternLayout>
        </Console>

        <RollingFile name="SERVICE"
                     fileName="${logDir}/${filenamePrefix}service.log"
                     filePattern="${logDir}/service/%d{yyyyMMdd}.%i.log">
            <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
            <PatternLayout>
                <pattern>${defaultPattern}</pattern>
            </PatternLayout>
            <Policies>
                <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
                <SizeBasedTriggeringPolicy size="${maxSize}"/>
            </Policies>
            <DefaultRolloverStrategy max="${maxCount}"/>
        </RollingFile>

        <RollingFile name="DEFAULT"
                     fileName="${logDir}/${filenamePrefix}default.log"
                     filePattern="${logDir}/default/%d{yyyyMMdd}.%i.log">
            <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
            <PatternLayout>
                <pattern>${defaultPattern}</pattern>
            </PatternLayout>
            <Policies>
                <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
                <SizeBasedTriggeringPolicy size="${maxSize}"/>
            </Policies>
            <DefaultRolloverStrategy max="${maxCount}"/>
        </RollingFile>

        <RollingFile name="RADIUS"
                     fileName="${logDir}/${filenamePrefix}radius.log"
```

```
            filePattern="${logDir}/radius/%d{yyyyMMdd}.%i.log">
    <ThresholdFilter level="DEBUG" onMatch="ACCEPT" onMismatch="DENY"/>
    <PatternLayout>
        <pattern>${defaultPattern}</pattern>
    </PatternLayout>
    <Policies>
        <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
        <SizeBasedTriggeringPolicy size="${maxSize}"/>
    </Policies>
    <DefaultRolloverStrategy max="${maxCount}"/>
</RollingFile>

<RollingFile name="GENERIC_AP"
            fileName="${logDir}/${filenamePrefix}generic-ap.log"
            filePattern="${logDir}/generic-ap/%d{yyyyMMdd}.%i.log">
    <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
    <PatternLayout>
        <pattern>${defaultPattern}</pattern>
    </PatternLayout>
    <Policies>
        <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
        <SizeBasedTriggeringPolicy size="${maxSize}"/>
    </Policies>
    <DefaultRolloverStrategy max="${maxCount}"/>
</RollingFile>

<RollingFile name="LEASE_SAVER"
            fileName="${logDir}/${filenamePrefix}lease_saver.log"
            filePattern="${logDir}/lease_saver/%d{yyyyMMdd}.%i.log">
    <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
    <PatternLayout>
        <pattern>${defaultPattern}</pattern>
    </PatternLayout>
    <Policies>
        <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
        <SizeBasedTriggeringPolicy size="${maxSize}"/>
    </Policies>
    <DefaultRolloverStrategy max="${maxCount}"/>
</RollingFile>

<RollingFile name="SHAPER"
            fileName="${logDir}/${filenamePrefix}shaper.log"
            filePattern="${logDir}/shaper/%d{yyyyMMdd}.%i.log">
    <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
    <PatternLayout>
        <pattern>${defaultPattern}</pattern>
    </PatternLayout>
    <Policies>
        <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
        <SizeBasedTriggeringPolicy size="${maxSize}"/>
    </Policies>
    <DefaultRolloverStrategy max="${maxCount}"/>
</RollingFile>

<RollingFile name="I18N"
            fileName="${logDir}/${filenamePrefix}i18n.log"
            filePattern="${logDir}/i18n/%d{yyyyMMdd}.%i.log">
    <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
    <PatternLayout>
        <pattern>${defaultPattern}</pattern>
    </PatternLayout>
    <Policies>
        <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
        <SizeBasedTriggeringPolicy size="${maxSize}"/>
    </Policies>
    <DefaultRolloverStrategy max="${maxCount}"/>
</RollingFile>

<RollingFile name="HAZELCAST"
            fileName="${logDir}/${filenamePrefix}hazelcast.log"
            filePattern="${logDir}/hazelcast/%d{yyyyMMdd}.%i.log">
```

```
            <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
            <PatternLayout>
                <pattern>${defaultPattern}</pattern>
            </PatternLayout>
            <Policies>
                <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
                <SizeBasedTriggeringPolicy size="${maxSize}"/>
            </Policies>
            <DefaultRolloverStrategy max="${maxCount}"/>
        </RollingFile>

<!--                leakDetectionThreshold    hikari-->

<!--        <RollingFile name="HIKARI" fileName="${logDir}/${filenamePrefix}hikariCP.log"-->
<!--                     filePattern="${logDir}/hazelcast/%d{yyyyMMdd}.%i.log">-->
<!--            <ThresholdFilter level="WARN" onMatch="ACCEPT" onMismatch="DENY"/>-->
<!--            <PatternLayout>-->
<!--                <pattern>${defaultPattern}</pattern>-->
<!--            </PatternLayout>-->
<!--            <Policies>-->
<!--                <SizeBasedTriggeringPolicy size="${maxSize}"/>-->
<!--            </Policies>-->
<!--            <DefaultRolloverStrategy max="${maxCount}"/>-->
<!--        </RollingFile>-->

        <Gelf name="Gelf"
              host="${gelfHost}"
              port="${gelfPort}"
              version="1.1"
              facility="eltex-pcrf"
              extractStackTrace="true"
              originHost="%host{fqdn}"
              maximumMessageSize="8192">
            <Field name="thread" pattern="%t"/>
            <Field name="level" pattern="%level"/>
            <Field name="severity" pattern="%-5level"/>
            <Field name="logger" pattern="%logger{1}"/>
            <Field name="location" pattern="%C{1}.%M(line:%L)"/>
        </Gelf>
    </Appenders>

    <Loggers>
<!--        Hikari logger    -->
<!--        <Logger name="com.zaxxer.hikari" level="WARN" additivity="false">-->
<!--            <appender-ref ref="HIKARI"/>-->
<!--        </Logger>-->

        <Logger name="org.eltex.softwlc.pcrf">
            <AppenderRef ref="SERVICE"/>
        </Logger>
        <Logger name="org.eltex.softwlc.pcrf">
            <AppenderRef ref="SERVICE"/>
        </Logger>
        <Logger name="io.vertx">
            <AppenderRef ref="SERVICE"/>
        </Logger>
        <Logger name="com.hazelcast"
                additivity="false">
            <AppenderRef ref="HAZELCAST"/>
        </Logger>

        <!-- Turn off debug messages about cluster from MongoDB driver -->
        <Logger name="org.mongodb.driver.cluster"
                level="INFO"/>

        <!-- This Logger writes only to radius.log -->
        <Logger name="org.eltex.softwlc.pcrf.vertx.radius"
                additivity="false">
            <AppenderRef ref="RADIUS"/>
        </Logger>
        <!-- This Logger also writes to radius.log -->
```

```
        <Logger name="net.jradius">
            <AppenderRef ref="RADIUS"/>
        </Logger>

        <!-- GenericAP Registrar logging -->
        <Logger name="org.eltex.softwlc.pcrf.vertx.radius.verticle.option82"
                additivity="false">
            <AppenderRef ref="GENERIC_AP"/>
        </Logger>

        <!-- This Logger writes only to lease_saver.log -->
        <Logger name="org.eltex.softwlc.pcrf.vertx.dhcp"
                additivity="false">
            <AppenderRef ref="LEASE_SAVER"/>
        </Logger>

        <!-- This Logger writes only to shaper.log -->
        <Logger name="org.eltex.softwlc.pcrf.vertx.shaper"
                additivity="false">
            <AppenderRef ref="SHAPER"/>
        </Logger>

        <Logger name="org.eltex.softwlc.pcrf.language"
                additivity="false">
            <AppenderRef ref="I18N"/>
        </Logger>

        <Root level="DEBUG">
            <AppenderRef ref="Gelf" level="${gelfLevel}"/>
        </Root>
    </Loggers>
</Configuration>
```

- (.):

```
<Property name="maxSize" value="5MB"/>
```

- , :

```
<Property name="maxCount" value="7"/>
```

- :

```
<Property name="logDir" value="/var/log/eltex-pcrf"/>
```

- Graylog (, ,):

```
<Property name="gelfHost" value="udp:lab3-test.eltex.loc"/>
<Property name="gelfPort" value="12201"/>
<Property name="gelfLevel" value="OFF"/
```

- :

```
<Root level="DEBUG">
```