

ActiveMQ

- HARD
-
-
-
- ActiveMQ

HARD

DHCP- RADIUS-. HARD MongoDB. HARD .

, ActiveMQ (Oracle AQ), . HARD ActiveMQ- .

ActiveMQ: /etc/hydra/activemq/instances-enabled/provisioning/activemq.xml

```

...
    <camelContext xmlns="http://camel.apache.org/schema/spring" id="camel">
        ...
        <route>
            <from uri="oracleTopic:topic:AIS_NET.HP_PROFILES_1?clientId=ActiveMQ& durableSubscriptionName=HP"
/>

            <multicast stopOnException="true">
                <to uri="activemq:queue:hydra_profiles_pppoe-rad"/>      <!-- Equipment profiles for HARD agent
at PPPoE RADIUS-server -->
                <to uri="activemq:queue:hydra_profiles_ran-dhcp"/>      <!-- Equipment profiles for HARD agent
at Radio Access Network DHCP-server -->
                <to uri="activemq:queue:hydra_profiles_ran-rad"/>      <!-- Equipment profiles for HARD agent
at Radio Access Network RADIUS-server -->
            </multicast>
        </route>
    </route>

    <from uri="oracleTopic:topic:AIS_NET.HP_EQUIPMENT_BINDS_1?clientId=ActiveMQ&
durableSubscriptionName=HP"/>
    <multicast stopOnException="true">
        <to uri="activemq:queue:hydra_equipment_binds_pppoe-rad"/>      <!-- Equipment binds for HARD
agent at PPPoE RADIUS-server -->
        <to uri="activemq:queue:hydra_equipment_binds_ran-dhcp"/>      <!-- Equipment binds for HARD
agent at Radio Access Network DHCP-server -->
        <to uri="activemq:queue:hydra_equipment_binds_ran-rad"/>      <!-- Equipment binds for HARD
agent at Radio Access Network RADIUS-server -->
    </multicast>
</route>
    ...
</camelContext>
...

```

HARD RADIUS- PPPoE: /etc/hydra/hard/hard.yml

```
...
#   ActiveMQ
syncer:
  stomp:
    #
    host: '192.168.123.45'
    #
    port: 61613
    #
    login: 'hydra'
    #
    password: 'Change me please'
  queues:
    #
    profiles: 'hydra_profiles_pppoe-rad'
    #
    binds: 'hydra_equipment_binds_pppoe-rad'
...
```

HARD RADIUS- Radio Access Network: /etc/hydra/hard/hard.yml

```
...
#   ActiveMQ
syncer:
  stomp:
    #
    host: '192.168.123.45'
    #
    port: 61613
    #
    login: 'hydra'
    #
    password: 'Change me please'
  queues:
    #
    profiles: 'hydra_profiles_ran-rad'
    #
    binds: 'hydra_equipment_binds_ran-rad'
...
```

: SNMP - . , «» «» BRAS'. .

HEX — .

ActiveMQ: /etc/hydra/activemq/instances-enabled/provisioning/activemq.xml

```
...
  <camelContext xmlns="http://camel.apache.org/schema/spring" id="camel">
    ...
    <route>
      <from uri="oracleTopic:topic:AIS_NET.HP_COMMANDS_1?clientId=ActiveMQ& durableSubscriptionName=HP"
/>
      <choice>
        <when>
          <xpath>count(/command/vc_command[contains(.,'/opt/hydra/scripts/reset-switch-port.sh')]) > 0</xpath>
          <to uri="activemq:queue:hydra_commands_reset-switch-port"/>
        </when>
        <otherwise>
          <to uri="activemq:queue:hydra_commands"/>
        </otherwise>
      </choice>
    </route>
    <route>
      <from uri="activemq:queue:hydra_command_results"/>
      <to uri="oracleTopic:topic:AIS_NET.HP_COM_RES_1?clientId=ActiveMQ& durableSubscriptionName=HP"/>
    </route>
    ...
  </camelContext>
...
```

HEX: /etc/hydra/hex/hex.yml

```
...
hydra:
  command_queues:
    # < >: < >
    - 'hydra_commands': 'hydra_command_results'
    - 'hydra_commands_reset-switch-port': 'hydra_command_results'
...
```

6.3+

vc_command , Base64. xpath , :

Bash

```
echo -n "your_command" | base64
```

```
<xpath>count(/command/vc_command[contains(.,'your_command_in_base64')]) > 0</xpath>
```

,

UTF-8	Base64
prev-ip-addr: next-ip-addr: 192.168.0.1	cHJldi1pcC1hZGRyOiAgbmV4dC1pcC1hZGRyOiAxOTluMTY4LjAuMQ==
prev-ip-addr: next-ip-addr: 192.168.0.2	cHJldi1pcC1hZGRyOiAgbmV4dC1pcC1hZGRyOiAxOTluMTY4LjAuMg==
prev-ip-addr: 192.168.0.1 next-ip-addr:	cHJldi1pcC1hZGRyOiAxOTluMTY4LjAuMSBuZXh0LWlwLWFkZHI6IA==
prev-ip-addr: 192.168.0.2 next-ip-addr:	cHJldi1pcC1hZGRyOiAxOTluMTY4LjAuMiBuZXh0LWlwLWFkZHI6IA==
prev-ip-addr:	cHJldi1pcC1hZGRyOg==

Base64 — cHJldi1pcC1hZGRyO, .

. RADIUS- — HARD . Oracle, .
- . Oracle,

AcitveMQ , . ActiveMQ <stop/>, :

ActiveMQ: /etc/hydra/activemq/instances-enabled/provisioning/activemq.xml

```
...
<camelContext xmlns="http://camel.apache.org/schema/spring" id="camel">
  <!-- AQ_BI_HYDRA_COMMANDS -->
  <route>
    <from uri="oracleTopic:topic:AIS_NET.HP_COMMANDS_1?clientId=ActiveMQ& durableSubscriptionName=HP"
  />
    <to uri="activemq:queue:hydra_commands_1"/>
  </route>
  <!-- AQ_BI_HYDRA_COM_RES -->
  <route>
    <from uri="activemq:queue:hydra_command_results_1"/>
    <to uri="oracleTopic:topic:AIS_NET.HP_COM_RES_1?clientId=ActiveMQ& durableSubscriptionName=HP" />
  </route>
  <!-- AQ_BI_HYDRA_PROFILES -->
  <route>
    <from uri="oracleTopic:topic:AIS_NET.HP_PROFILES_1?clientId=ActiveMQ& durableSubscriptionName=HP"
  />
    <stop/>
    <to uri="activemq:queue:hydra_profiles_1"/>
  </route>
  <!-- AQ_BI_HYDRA_EQUIP_BINDS -->
  <route>
    <from uri="oracleTopic:topic:AIS_NET.HP_EQUIPMENT_BINDS_1?clientId=ActiveMQ&
durableSubscriptionName=HP" />
    <stop/>
    <to uri="activemq:queue:hydra_equipment_binds_1"/>
  </route>
  <!-- - AQ_BI_HYDRA_MESSAGES -->
  <route>
    <from uri="oracleTopic:topic:AIS_NET.HYDRA_MESSAGES?clientId=ActiveMQ&
durableSubscriptionName=HP" />
    <stop/>
  </route>
</camelContext>
...
```

to : «» stop.
Oracle RETENTION ALL_QUEUES.

(.). ActiveMQ , (STOMP, AMPQ, REST),

AcitveMQ :

ActiveMQ: /etc/hydra/activemq/instances-enabled/provisioning/activemq.xml

```
...
<camelContext xmlns="http://camel.apache.org/schema/spring" id="camel">
  <route>
    <from uri="activemq:queue:hydra_input_messages_1"/>
    <to uri="oracleTopic:topic:AIS_NET.HB_SYS_IN_1?clientId=ActiveMQ& durableSubscriptionName=HB"/>
  </route>
</camelContext>
...
```

AIS_PROVISIONING :

sqlplus

```
BEGIN
  DBMS_AQADM.GRANT_QUEUE_PRIVILEGE(
    privilege => 'enqueue',
    queue_name => 'AIS_NET.HB_SYS_IN_1',
    grantee => 'AIS_PROVISIONING');
END;
```

ActiveMQ. , REST API:

REST API

```
echo -n "body=<system_message>
  <vc_message_type>ProvisioningCurrentStateRequest</vc_message_type>
  <n_firm_id>100</n_firm_id>
  <content>
    <vc_key>Shaper-IP</vc_key>
    <vc_value>127.0.0.1</vc_value>
    <n_value />
    <d_value />
  </content>
</system_message>" | curl -u admin:admin --data-binary '@-' "http://192.168.1.1:8161/api/message/hydra_input_messages_1?type=queue"
```

Shaper-IP - (IP-), 127.0.0.1 - , 192.168.1.1 - IP- ActiveMQ.

ActiveMQ

, ActiveMQ . ActiveMQ

, activemq.xml

ActiveMQ /etc/hydra/activemq/hydra/activemq.xml

```
...
<broker xmlns="http://activemq.apache.org/schema/core" brokerName="localhost" dataDirectory="${activemq.
data}" useJmx="false" advisorySupport="false">

    <!--
    Use VM cursor
    For more information, see:
    http://activemq.apache.org/message-cursors.html
    -->
    <destinationPolicy>
        <policyMap>
            <policyEntries>
                <policyEntry topic="" producerFlowControl="true" memoryLimit="1000mb">
                    <pendingSubscriberPolicy>
                        <vmCursor />
                    </pendingSubscriberPolicy>
                </policyEntry>
                <policyEntry queue="" producerFlowControl="true" memoryLimit="1000mb">
                    <pendingQueuePolicy>
                        <vmQueueCursor/>
                    </pendingQueuePolicy>
                </policyEntry>
            </policyEntries>
        </policyMap>
    </destinationPolicy>
...

```

ActiveMQ /etc/hydra/activemq/hydra/activemq.xml

```
...
<broker xmlns="http://activemq.apache.org/schema/core" brokerName="localhost" dataDirectory="${activemq.
data}">

    <destinationPolicy>
        <policyMap>
            <policyEntries>
                <policyEntry topic="" >
                    <!-- The constantPendingMessageLimitStrategy is used to prevent
                    slow topic consumers to block producers and affect other consumers
                    by limiting the number of messages that are retained
                    For more information, see:

                    http://activemq.apache.org/slow-consumer-handling.html

                    -->
                    <pendingMessageLimitStrategy>
                        <constantPendingMessageLimitStrategy limit="1000"/>
                    </pendingMessageLimitStrategy>
                </policyEntry>
            </policyEntries>
        </policyMap>
    </destinationPolicy>
...

```