This article describes ACE health monitoring (probes), how to configure it, and how to troubleshoot issues with probes that you may encounter.

Contents

Overview of ACE Health Monitoring

This section describes health monitoring on the ACE. The ACE uses probes that you configure to track the state of a server. By default, no probes are configured in the ACE. Also referred to as out-of-band (OOB) health monitoring, the ACE verifies the server response to a probe or checks for any network problems that can prevent a client from reaching a server. Based on the server response, the ACE can place the server in or out of service and can make reliable load-balancing decisions.

You can also use health monitoring to detect failures for a gateway or a host in high-availability (redundant) configurations. For more information, see the *Cisco Application Control Engine Module Administration Guide*.

The ACE evaluates the health of a server by marking the probes as follows:

- Passed?The server returns a valid response.
- Failed?The server fails to provide a valid response to the ACE and the ACE is unable to reach a server for a specified number of retries.

By configuring the ACE for health monitoring, the ACE sends active probes periodically to determine the server state. The ACE supports 4096 unique probe configurations, which includes ICMP, TCP, HTTP, and other predefined health probes. The ACE can execute only up to 200 concurrent scripted probes at a time. The ACE also allows the opening of 2048 sockets simultaneously.

You can associate the same probe with multiple real servers or server farms. Each time that you use the same probe again, the ACE counts it as another probe instance. You can allocate a maximum of 16,000 probe instances.

Configuring Probes

You can configure health probes on the ACE to actively make connections and explicitly send traffic to servers. The probes determine whether the health status of a server passes or fails by the server's response.

Configuring active probes is a three-step process:

1. Configure the health probe with a name, type, and attributes.

2. Associate the probe with one of the following:

    ◊ A real server.

    ◊ A real server and then associate the real server with a server farm. You can associate a single

probe or multiple probes with real servers within a server farm.

◊ A server farm. All servers in the server farm receive probes of the associated probe types.

3. Place the real server or server farm in service.

Example of a Probe Configuration

The following example shows a running configuration that load balances DNS traffic across multiple real servers and transmits and receives UDP data that spans multiple packets. The configuration uses a UDP health probe.

```
access-list ACL1 line 10 extended permit ip any any

probe udp UDP
  interval 5
  passdetect interval 10
  description THIS PROBE IS INTENDED FOR LOAD BALANCING DNS TRAFFIC
  port 53
  send-data UDP_TEST

rserver host SERVER1
  ip address 192.168.10.45
  inservice
rserver host SERVER2
  ip address 192.168.10.46
  inservice
rserver host SERVER3
  ip address 192.168.10.47
  inservice

serverfarm host SFARM1
  probe UDP
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER3
    inservice

class-map match-all L4UDP-VIP_114:UDP_CLASS
   2 match virtual-address 192.168.120.114 udp eq 53

policy-map type loadbalance first-match L7PLBSF_UDP_POLICY
  class class-default
    serverfarm SFARM1
policy-map multi-match L4SH-Gold-VIPs_POLICY
    class L4UDP-VIP_114:UDP_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_UDP_POLICY
    loadbalance vip icmp-reply
    nat dynamic 1 vlan 120
    connection advanced-options 1SECOND-IDLE

interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.70 netmask 255.255.255.0 pat
```

```
  service-policy input L4SH-Gold-VIPs_POLICY
  no shutdown

ip route 10.1.0.0 255.255.255.0 192.168.120.254
```

Troubleshooting ACE Health Monitoring

This section describes how to troubleshoot two common probe configuration issues.

Troubleshooting an HTTP Probe Error

In this first scenario, you have configured an HTTP probe, but the real server's health status is displayed as FAILED and the Last disconnect err field indicates that an invalid status code was received as displayed in the **show probe detail** command output. You have checked your server and it is up and running. A packet capture on the server also shows that everything is fine. Where is the issue?

1. Display the probe status details by entering the following command:

```
ACE_module5/Admin# show probe detail

 probe       : HTTP_PROBE
 type        : HTTP
 state       : ACTIVE
 description :
----------------------------------------------
   port      : 80      address     : 0.0.0.0         addr type  : -
   interval  : 10      pass intvl  : 10              pass count : 3
   fail count: 3       recv timeout: 10
   http method      : GET
   http url         : /
   conn termination : GRACEFUL
   expect offset    : 0         , open timeout     : 1
   expect regex     : -
   send data        : -
                ------------------ probe results ------------------
   associations ip-address      port  porttype probes   failed   passed   health
   ------------ ---------------+-----+--------+--------+--------+--------+------
   rserver     : SERVER1
                 192.168.10.45   80    --       2        2        0        FAILED

   Socket state        : CLOSED
   No. Passed states   : 0         No. Failed states : 1
   No. Probes skipped  : 0         Last status code  : 200 <------- Last status code from server
   No. Out of Sockets  : 0         No. Internal error: 0
   Last disconnect err : Received invalid status code <-------
   Last probe time     : Tue Apr  7 16:17:26 2009
   Last fail time      : Tue Apr  7 16:17:16 2009
   Last active time    : Never
```

The Last disconnect err field indicates that the ACE received an invalid status code. This error means that you have has not configured the **expect status** command for the probe.

2. Confirm this finding by entering the following command:

```
ACE_module5/Admin# show running-config probe
Generating configuration....

 probe http HTTP_PROBE
```

```
   interval 10
   passdetect interval 10
   open 1
```

3. Correct the problem by entering the following commands:

```
ACE_module5/Admin# config
Enter configuration commands, one per line.  End with CNTL/Z.
ACE_module5/Admin(config)# probe http HTTP_PROBE
ACE_module5/Admin(config-probe-http)# expect status 200 200 <------- 200 indicates the 200 OK mess
ACE_module5/Admin(config-probe-http)# end
```

4. Confirm the configuration by entering the following command:

```
ACE_module5/Admin# show running-config probe
Generating configuration....

 probe http HTTP_PROBE
   interval 10
   passdetect interval 10
   expect status 200 200
   open 1
```

5. Display the probe status details again and observe that the server health status value is SUCCESS by entering the following command:

```
ACE_module5/Admin# show probe HTTP_PROBE detail

 probe       : HTTP_PROBE
 type        : HTTP
 state       : ACTIVE
 description :
----------------------------------------------
   port      : 80      address     : 0.0.0.0         addr type  : -
   interval  : 10      pass intvl  : 10              pass count : 3
   fail count: 3       recv timeout: 10
   http method      : GET
   http url         : /
   conn termination : GRACEFUL
   expect offset    : 0         , open timeout     : 1
   expect regex     : -
   send data        : -
               ------------------- probe results ------------------
   associations ip-address      port  porttype probes   failed   passed   health
   ------------ ---------------+-----+--------+--------+--------+--------+------
   rserver     : SERVER1
                 192.168.10.45   80    --       24       15       9        SUCCESS

   Socket state        : CLOSED
   No. Passed states   : 1         No. Failed states : 1
   No. Probes skipped  : 0         Last status code  : 200
   No. Out of Sockets  : 0         No. Internal error: 0
   Last disconnect err :  -  <------- No error indicated now. The probe is successful.
   Last probe time     : Tue Apr  7 16:21:05 2009
   Last fail time      : Tue Apr  7 16:17:16 2009
   Last active time    : Tue Apr  7 16:20:05 2009
```

Troubleshooting an HTTPS Probe Error

In addition to the methods for Troubleshooting an HTTP Probe Error, use SSL statistics to troubleshoot HTTPS probe failures. HTTPS probe traffic runs in the Admin virtual context so view the output of the show stats crypto client command in that context.

Troubleshooting an SNMP Probe Issue

In this scenario, you have configured an SNMP probe, but the Last disconnect err field indicates that the sum of the weights does not add up to the maximum weight value as displayed in the output of the **show probe detail** command.

1. Display the probe status details by entering the following command:

```
ACE_module5/test# show probe detail

 probe       : SNMP_PROBE
 type        : SNMP
 state       : ACTIVE
 description : snmp probe
----------------------------------------------
   port      : 161     address     : 0.0.0.0         addr type  : -
   interval  : 15      pass intvl  : 10              pass count : 3
   fail count: 3       recv timeout: 10
   version   : 2c      community   : test_comm
   oid string #1 : .1.3.6.1.2.1.4.3.0
   type      : ABSOLUTE        max value : 1000000000
   weight    : 10000           threshold : 1000000000
               ------------------ probe results ------------------
   associations ip-address      port  porttype probes   failed   passed   health
   ------------ ---------------+-----+--------+--------+--------+--------+------
   serverfarm  : least-loaded, predictor least-loaded
     real      : SERVER1[0]
                 192.168.10.45   161   --       0        0        0        INIT

   Socket state        : CLOSED
   No. Passed states   : 0         No. Failed states : 0
   No. Probes skipped  : 0         Last status code  : 0
   No. Out of Sockets  : 0         No. Internal error: 30
   Last disconnect err : Sum of weights don't add up to max weight value <------- Error condition
   Last probe time     : Never
   Last fail time      : Never
   Last active time    : Never
   Server load         : 16000 <------- Note the server load value
```

The reason for this error is that the **weight** command needs to be configured when you have multiple OIDs configured for a single probe and from those OIDs if you want to give priority to a specific OID.

The sum of the weights should equal 16000 (see the Server Load field). For a single OID, the **weight** command does not have any significance.

2. Display the probe configuration by entering the following command:

```
ACE_module5/Admin# show running-config probe

probe snmp SNMP_PROBE
  description snmp probe
  port 161
  interval 15
```

```
  passdetect interval 10
  version 2c
  community TEST_COMM
  oid .1.3.6.1.2.1.4.3.0
    type absolute max 1000000000
    weight 10000 <-------
```

In the above configuration, the weight is configured as 10000 for a single OID. The ACE is expecting another OID to be configured in the probe and the sum of both weights should equal 16000.

The configuration is not complete and the ACE is expecting additional parameters in the probe configuration. Because there is not another OID in the configuration, the ACE is not able to calculate the load and that is why the "Sum of weights don't add up to max weight value" error message appears.

3. Resolve the issue by modifying the probe configuration as follows:

```
probe snmp SNMP_PROBE
  description test
  port 161
  interval 15
  passdetect interval 60
  version 2c
  community test_comm
  oid .1.3.6.1.2.1.4.3.0
    type absolute max 1000000000
    weight 10000
  oid .1.3.6.1.2.1.4.10.0
    type absolute max 1000000000
    weight 6000 <------- 10000 + 6000 = 16000
```

4. Display the probe status details again by entering the following command:

```
ACE_module5/test# show probe SNMP_PROBE detail

 probe       : snmp1
 type        : SNMP
 state       : ACTIVE
 description : snmp probe
----------------------------------------------
   port      : 161     address     : 0.0.0.0         addr type  : -
   interval  : 15      pass intvl  : 10              pass count : 3
   fail count: 3       recv timeout: 10
   version   : 2c      community   : test_comm

   oid string #1 : .1.3.6.1.2.1.4.3.0
   type      : ABSOLUTE        max value : 1000000000
   weight    : 10000           threshold : 1000000000

   oid string #2 : .1.3.6.1.2.1.4.10.0
   type      : ABSOLUTE        max value : 1000000000
   weight    : 6000            threshold : 1000000000
               ------------------ probe results ------------------
   associations ip-address      port  porttype probes   failed   passed   health
   ------------ ---------------+-----+--------+--------+--------+--------+------
   serverfarm  : least-loaded, predictor least-loaded
     real      : SERVER1[0]
               192.168.10.45   161   --       4143     0        4143     SUCCESS

   Socket state        : CLOSED
   No. Passed states   : 1         No. Failed states : 0
   No. Probes skipped  : 0         Last status code  : 0
```

```
   No. Out of Sockets  : 0         No. Internal error: 0
   Last disconnect err :  -  <-------  No error indicated now. The probe is successful.
   Last probe time     : Mon Apr  6 09:12:54 2009
   Last fail time      : Never
   Last active time    : Sun Apr  5 15:57:28 2009
   Server load         : 0
```

Using the Last Status Code Field

Details regarding the last status code field can be provided for nontrivial probes. For example, in the case of scripted probe PROBENOTICE_PROBE, the status code 30001 means that the probe is successful and the value 30002 indicates an error in probe arguments. The last disconnect error for status code 30002 displays "Did not receive correct response from the server," but the actual issue is related to arguments in the probe configuration, which can be checked by looking at the script for the probe.

```
ACE_module5/Admin# show probe TEST detail

 probe       : TEST
 type        : SCRIPTED
 state       : ACTIVE
 description :
----------------------------------------------
   port      : 0       address     : 0.0.0.0         addr type  : -
   interval  : 15      pass intvl  : 20              pass count : 3
   fail count: 3       recv timeout: 10
   script filename  : PROBENOTICE_PROBE
                       ---------------------- probe results ---------------------
   probe association   probed-address  probes     failed     passed     health
   ------------------- ---------------+----------+----------+----------+-------
   serverfarm  : sf1
     real      : rs2[0]
                       23.0.0.5        4082       54         4028       SUCCESS

   Socket state        : RESET
   No. Passed states   : 6         No. Failed states : 5
   No. Probes skipped  : 8         Last status code  : 30001 <------- Indicates success
   No. Out of Sockets  : 0         No. Internal error: 0
   Last disconnect err :  -
   Last probe time     : Wed Apr  8 04:44:41 2009
   Last fail time      : Tue Apr  7 12:02:10 2009
   Last active time    : Tue Apr  7 12:03:45 2009
```