

Configure An External Load Balancer

This document describes how to configure an external load balancer to handle communications with Clustered AtScale.

Before You Begin

- ▲ Obtain the hostnames of the nodes that will be included in your AtScale cluster.

About This Task

You must configure an external load balancer to handle queries from client business intelligence (BI) applications such as Tableau Desktop and Microsoft Excel, requests from browsers which are connected to the Design Center, and communication between AtScale applications. Your load balancer can be a software (for example, HAProxy) or a hardware (for example, F5 or Cisco) appliance.

Procedure

Load Balancer Port Preparation

The following ports need to be load balanced between the AtScale hosts. In addition, note the following restrictions:

- ▲ Your load balancer must support websockets
- ▲ A round robin configuration with IP address session affinity (that is, "ip address sticky sessions") is strongly recommended. Running without session affinity will significantly degrade performance.

Service	Frontend Port	Transport Mode	Backend Port	Health Check
Design Center	10500	http**	10500	HTTP GET /ping
Engine HTTP	10502	http	10502	HTTP GET /health
Authentication	10503	http	10503	(None)
Engine Thrift *	1111	tcp	1111	HTTP GET :10502/health
Query Engine HTTP	10541	http	10541	HTTP GET /health

Query Engine Thrift *	11111	tcp	11111	HTTP GET :10502/health
Query Engine pgwire***	15432	tcp	15432	HTTP GET :10502/health

*If additional AtScale Environments are configured, the engine will open ports based on those settings. These new ports will need to be added to your load balancer.

**Must support websockets. Some load balancers may not support websockets in http mode, if this is the case try tcp mode.

***You may encounter issues when connecting to the `pgwire` endpoint from HA AtScale configurations, which may require additional configuration to resolve. If this occurs, contact AtScale Support.



Note: Some load balancers have problems handling some forms of http traffic. If you experience problems with your load balancer you may have to use the tcp transport mode.

Table 1. Default Load Balancer Values

Property	Suggested Value
timeout connect	5000ms
timeout client	130m
timeout server	130m
timeout tinnel (websockets)	3600s

Example Configuration For HAProxy 1.7

AtScale provides a script, `generate-haproxy-cfg.sh`, for you to generate an example HAProxy 1.7 load balancer configuration. For information, see [HAProxy documentation](#).

Steps To Generate An Example HAProxy 1.7 Configuration:

1. Log in to any one of the nodes that you will use for your AtScale cluster.

The AtScale installer package must already be installed, but AtScale does not need to be configured and running.

2. Run the following command as the `atscale` user, specifying only the hostnames of the AtScale Application nodes as the value of the `-ah` (AtScale hostnames) parameter:

```
/opt/atscale/versions/<package_version>/bin/installer/generate-haproxy-cfg.sh -ah '<comma-delimited string of AtScale Application host names>'
```

Results

The script generates a file at `/opt/atscale/conf/haproxy.cfg.sample`. You can either install this configuration in HAProxy 1.7 or use it as a reference configuration.

▲ Example:

```
#####
# This is an example HAProxy configuration file, tested with HAProxy 1.7, demonstrating
# how the AtScale services should be configured when AtScale is fronted with an external
# HAProxy 1.7 load balancer. You must adapt these settings to the load balancer you are using.
#
# HAProxy config docs: https://cbonte.github.io/haproxy-dconv/1.7/configuration.html
#####

####
# HAProxy Defaults
####
global
    maxconn 256

defaults
    option forwardfor except 127.0.0.1
    mode http
    timeout connect 5000ms
    timeout client 130m
    timeout server 130m
    # timeout tunnel needed for websockets
    timeout tunnel 3600s
    default-server init-addr last,libc,none

####
# AtScale Service Frontends
####
frontend design_center_front
    bind *:10500
    default_backend design_center_back
frontend engine_http_front
    bind *:10502
    default_backend engine_http_back
frontend auth_front
    bind *:10503
    default_backend auth_back

####
```

```

# AtScale Engine Frontends
####
frontend engine_tcp_front_11111
  mode tcp
  bind *:11111
  default_backend engine_tcp_back_11111
frontend engine_tcp_front_11112
  mode tcp
  bind *:11112
  default_backend engine_tcp_back_11112
frontend engine_tcp_front_11113
  mode tcp
  bind *:11113
  default_backend engine_tcp_back_11113

####
# AtScale Service Backends
####
backend design_center_back
  option httpchk GET /ping HTTP/1.1\r\nHost:\ www
  http-check expect status 200
  server atscale-node-01.atscale.com atscale-node-01.atscale.com:10500 check
  server atscale-node-02.atscale.com atscale-node-02.atscale.com:10500 check
  server atscale-node-03.atscale.com atscale-node-03.atscale.com:10500 check
backend engine_http_back
  option httpchk GET /health HTTP/1.1\r\nHost:\ www
  http-check expect status 200
  server atscale-node-01.atscale.com atscale-node-01.atscale.com:10502 check
  server atscale-node-02.atscale.com atscale-node-02.atscale.com:10502 check
  server atscale-node-03.atscale.com atscale-node-03.atscale.com:10502 check
backend auth_back
  server atscale-node-01.atscale.com atscale-node-01.atscale.com:10503 check
  server atscale-node-02.atscale.com atscale-node-02.atscale.com:10503 check
  server atscale-node-03.atscale.com atscale-node-03.atscale.com:10503 check

####
# AtScale Engine Backends
####
backend engine_tcp_back_11111
  mode tcp
  option httpchk GET /health HTTP/1.1\r\nHost:\ www
  http-check expect status 200
  server atscale-node-01.atscale.com atscale-node-01.atscale.com:11111 check port 10502
  server atscale-node-02.atscale.com atscale-node-02.atscale.com:11111 check port 10502
  server atscale-node-03.atscale.com atscale-node-03.atscale.com:11111 check port 10502
backend engine_tcp_back_11112
  mode tcp
  option httpchk GET /health HTTP/1.1\r\nHost:\ www
  http-check expect status 200
  server atscale-node-01.atscale.com atscale-node-01.atscale.com:11112 check port 10502
  server atscale-node-02.atscale.com atscale-node-02.atscale.com:11112 check port 10502
  server atscale-node-03.atscale.com atscale-node-03.atscale.com:11112 check port 10502
backend engine_tcp_back_11113
  mode tcp
  option httpchk GET /health HTTP/1.1\r\nHost:\ www
  http-check expect status 200
  server atscale-node-01.atscale.com atscale-node-01.atscale.com:11113 check port 10502
  server atscale-node-02.atscale.com atscale-node-02.atscale.com:11113 check port 10502
  server atscale-node-03.atscale.com atscale-node-03.atscale.com:11113 check port 10502

```

Terminating TLS At An External Load Balancer

AtScale 2020.3.1 introduces the ability to terminate TLS at the external load balancer. Terminating TLS at the external load balancer frees your backend servers from the compute-intensive work of encrypting and decrypting all of your traffic and allows certificate management to reside in the load balancer. Follow the steps below to terminate TLS at an external load balancer.

Procedure

1. Setup AtScale in insecure mode (TLS-disabled).
2. Create a new Virtual IP Address that is TLS-enabled (see [ports](#)).
3. Configure the AtScale UI with the new TLS-enabled VIP address. Via the insecure VIP, navigate to the **Engine Settings (Settings > Organization Settings: Engine)** page in the Design Center and update the following settings to reflect the TLS-enabled VIP:
 1. `load.balancer.mdx.url` : Base URL to reach the engine's MDX endpoint via the customer's load balancer. Must include the full HTTPS protocol and port. An example URL is as follows: `https://secure-lb.atscale.com:10502`
 2. `load.balancer.sql.url` : Base URL to reach the engine's SQL endpoint via the customer's load balancer. Must include the full HTTPS protocol and port. An example URL is as follows: `https://secure-lb.atscale.com:11111`
 3. `load.balancer.pgsql.url` : Base URL for the engine's PostgreSQL-compatible SQL endpoint. This should be in the format `schema://hostname:port` .



Note: Use http or https instead of JDBC protocol. AtScale interprets the protocol for publication.

4. Navigate to the **Global Settings - Configuration (Settings > Global Settings: Configuration)** page and update the following settings to reflect the TLS-enabled VIP.
 1. **Application Load Balancer URL** and **Engine Load Balancer URL**.
 - ▲ An example Application Load Balancer URL is: `https://secure-lb.atscale.com:10500` .
 - ▲ An example Engine Load Balancer URL is `https://secure-lb.atscale.com:10502` .



Note: If the user updates the values from the global options page, it will be the same for all the organizations on your environment. If the settings are updated on the organization level, it will only be updated specific to that organization.

5. From your external load balancer, only allow traffic from internal IPs for the insecure virtual IP of your load balancer.
6. **Optional:** In the event that there are errors with the TLS, allow administrators' IPs to connect to the insecure VIP.