LEOSTREAM™

Leostream Platform – Scalability Guide

Manage user connections to anything – anytime, anywhere, from any device

Version 9.1
October 2021

Contacting Leostream

Leostream Corporation
271 Waverley Oaks Rd
Suite 206
Waltham, MA 02452
USA

http://www.leostream.com

Telephone: +1 781 890 2019

To submit an enhancement request, email features@leostream.com.
To request product information or inquire about our future directions, email sales@leostream.com.

Copyright

Trademarks

The following are trademarks of Leostream Corporation.
Leostream™
The Leostream graphical logo™

The absence of a product name or logo from this list does not constitute a waiver of the trademark or other intellectual property rights concerning that product, name, or logo by Leostream.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. UNIX is a registered trademark of The Open Group. OpenLDAP is a trademark of The OpenLDAP Foundation. Microsoft, Active Directory, Hyper-V, Windows, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other brand and product names are trademarks or registered trademarks of their respective holders. Leostream claims no right to use of these marks.

Patents

Leostream software is protected by U.S. Patent 8,417,796.

Contents

Contents 3
Considerations for Production Deployments 4
Using Clusters to Maximize Availability 5
  Benefits of Using a Cluster 5
  Creating a Cluster 5
  Using the Cluster Management Page 8
  Removing Connection Brokers from a Cluster 10
  Updating Connection Broker Clusters to New Versions 10
  Spreading a Cluster across Multiple Datacenters/Regions 10
  Managing Different Clusters in Different Data Centers 12
  Building a Cluster for Disaster Recovery 14
Distributing User Logins and Network Traffic 14
  Citrix™ NetScaler™ Setup 15
  F5® BIG-IP® Load Traffic Manager™ (LTM) Setup 16
  Connection Broker User Redirection 16
    Example: User Redirection Scenario 16
    Setting up User Redirection in the Connection Broker 17
    Applying Forwarding only to Users that are already Assigned Desktops 19
Using an External Database 20
  Sizing the External Database 20
    Database Space Requirements 20
    Database Transaction Requirements 20
    Removing Deleted Database Records 21
  Switching to an External Database 22
  Database Failover 24
    Specifying a Failover Database 24
    Using Microsoft SQL Server Always On Availability Groups 25
Appendix A: Leostream Network Architecture 26

Considerations for Production Deployments

Desktop deployment is mission critical to many businesses. As such, you want to scale your Connection Broker deployment in a manner that ensures:

- Availability
- Disaster Recovery
- Capacity

Availability and *disaster recovery* ensure that your users are always able to log in through the Connection Broker. To achieve high availability, you must ensure that if a Connection Broker fails, another broker is available to handle connections. For disaster recovery, you must ensure that, if an entire datacenter goes down, users are able to log in to resources in a disaster recovery datacenter.

Capacity describes the number of users that can simultaneously log into your Connection Broker with reasonable latency. It is possible to design your Connection Broker deployment to have high availability, while still having capacity issues.

To accomplish these goals in a production-class environment, create systems that ensure the redundancy, resiliency, and scalability of your deployment, including:

- Create a Connection Broker cluster with sufficient Connection Brokers to handle user logins in the event that a server hosting one of the Connection Broker fails. For added resiliency ensure that you place individual Connection Brokers on different servers.

- Integrate with global and local load balancers, to optimize Connection Broker performance.

- Establish a schedule for backing up your Connection Broker database. Implement your site standard database backup procedure, to ensure that your data is protected.

- Create weekly snapshots of each Connection Broker virtual machine. By backing up the entire Connection Broker virtual machine, you do not need a separate backup procedure for the underlying Connection Broker operating system.

- Create monthly clones of each Connection Broker virtual machine. Leostream recommends storing these backups in an off-site location. Test your restore process to ensure that the media can be read, and that procedures are correctly documented.

- Use DNS to configure your Connection Broker IP addresses. (See the Leostream DNS Setup Guide)

- Never perform a Connection Broker upgrade without first taking a snapshot of your existing Connection Broker virtual machine. Always test upgrades in an isolated deployment, before rolling out to your production environment.

- Minimize the latency between your Connection Broker cluster and its external database, as well as the latency between your Connection Broker cluster and your authentication servers.

Using Clusters to Maximize Availability

A Connection Broker *cluster* is a group of Connection Brokers that share the same PostgreSQL, Azure SQL, or Microsoft SQL Server® database. A common cluster uses three to five Connection Brokers.

 A cluster cannot contain a mixture of version 8 and version 9 Connection Brokers.

Benefits of Using a Cluster

Clusters address the three scalability goals, as follows:

- **Availability:** Using clusters enhances availability by allowing any Connection Broker instance to handle the necessary system functions without operator intervention. If one Connection Broker in the cluster fails, user logins are processed by the other Connection Brokers, resulting in no break in the end-user experience. Connection Broker instances that are not handling logins automatically process other system tasks.
- **Disaster Recovery:** Using clusters also allows you to mitigate system or site failures. Run each Connection Broker in the cluster on a different virtualization host, to ensure resiliency to a host failure. Place Connection Brokers or entire clusters in different datacenters or regions, to support disaster recovery scenarios.
- **Capacity:** The number of logins per second that can be handled depends on the overall structure of your Connection Brokers, database, and authentication server. Typically, each Connection Broker can handle five logins per second. To increase this throughput, add additional Connection Brokers on different hosts and spread the traffic between the Connection Brokers using a load balancer. The throughput scales linearly when using up to ten Connection Brokers.

  If the authentication server infrastructure cannot handle the load, the Connection Broker buffers login requests and the login time climbs quickly. After two minutes, the login requests time out and the user must log in again.

Creating a Cluster

To create a cluster of Connection Broker:

1. Install a standalone Connection Broker. By default, the Connection Broker uses an internal database.

    Because Connection Brokers run within virtual machines, their performance varies according to the overall load on that host, in addition to the load on the particular Connection Broker. Ensure that your Connection Brokers have sufficient resources on your virtualization host.

2. Apply your Leostream license to this Connection Broker. See “Entering Your License” in Chapter 2 of the Leostream Administrator’s Guide for the complete procedure.

3. Optionally configure this Connection Broker with centers, pools, authentication servers, etc. At this point, any information you enter into the Connection Broker is stored in its internal database. Often, at this stage, you are working on a proof-of-concept for your deployment.

4. To begin building a Connection Broker cluster, first obtain the address and credentials for a PostgreSQL, Azure SQL, or Microsoft SQL Server database server. You must connect all the Connection Broker in you cluster to the same database.

    If using Azure SQL, you must create the database prior to connecting the Connection Broker. For PostgreSQL and Microsoft SQL Server, Leostream can create a new database if one does not already exist.

5. To connect the first Connection Broker to the external database, go to the Connection Broker **> System > Maintenance** page.

6. Select one of the options to switch to an external database and click **Next**.

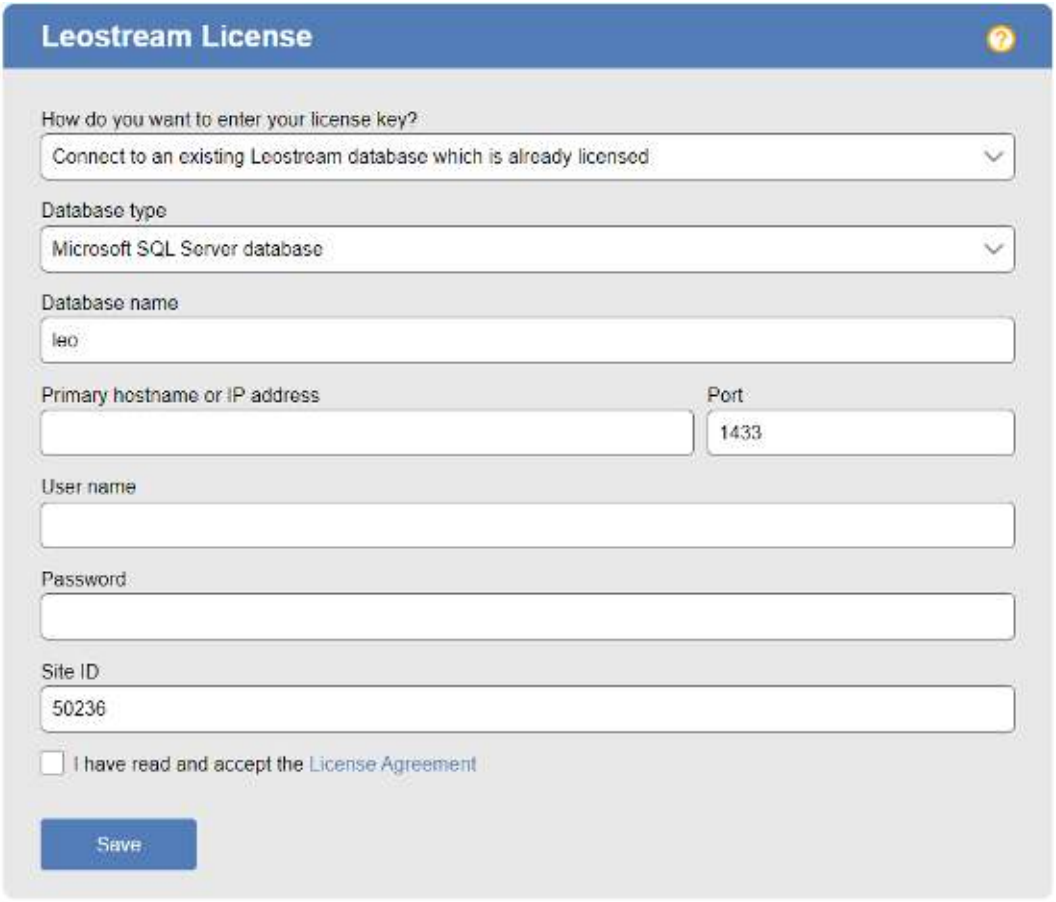
7. In the **Database** form, switch this Connection Broker to the new external database. When switching the database, note the database name and Site ID for this Connection Broker. See Switching to an External Database for complete instructions.

   When you switch your first Connection Broker over to an empty external database, the Connection Broker automatically populates the database with the information currently stored in the Connection Broker internal database.

8. To add additional Connection Brokers to the cluster, install individual Connection Brokers virtual appliances on different virtualization hosts. These Connection Brokers can be located in any data center, as long as the Connection Broker can communicate with your database server.

9. For each additional Connection Brokers, log into the Connection Broker as the default administrator. The Leostream License form opens.

10. Select **Connect to an existing Leostream database which is already licensed** from the **How do you want to enter your license key** drop-down menu. The form updates as shown in the following figure.



11. Select the type of database you will connect to from the **Database type** drop-down menu.

12. In the remaining fields, enter the information used to switch the original Connection Broker to the external database.

13. Select the **I have read and accept the License Agreement** checkbox.

14. Click **Save**.

All Connection Brokers in the cluster work off of a common job queue. When a new Connection Broker is added to the cluster, a `heartbeat` job for that Connection Broker appears in the **> System > Job Queue** page. This heartbeat job checks the Connection Broker status every five minutes, and is used to monitor the status of each Connection Broker when collecting Connection Broker Metrics and when reporting Connection Broker status on the **> System > Cluster Management** page.

Using the Cluster Management Page

The **> System > Cluster Management** page lists the Connection Brokers in the cluster and their characteristics. You can modify the order and type of characteristics displayed on this page by clicking the **Customize columns** link at the top-right side of the page.

You can display any or all of the following characteristics.

Actions

Links indicating the actions you can perform on a particular Connection Broker, including:

- **Remove:** Removes this Connection Broker from the cluster. You can remove a Connection Broker only if its status is `Unavailable` or `Stopped`.

Name

The Connection Broker virtual appliance hostname, by default, `leostream`.

IP Address

The Connection Broker IP address, as entered into the **Bridged** interface in the **> System > Network** page.

Status

Indicates the availability of each Connection Broker for processing jobs in the job queue. Possible status values are as follows.

- **Running:** Indicates this Connection Broker is running and available to process jobs in the job queue.

- **Stopped:** Indicates the `heartbeat` job associated with this Connection Broker has been cancelled. A stopped Connection Broker cannot process jobs in the job queue.

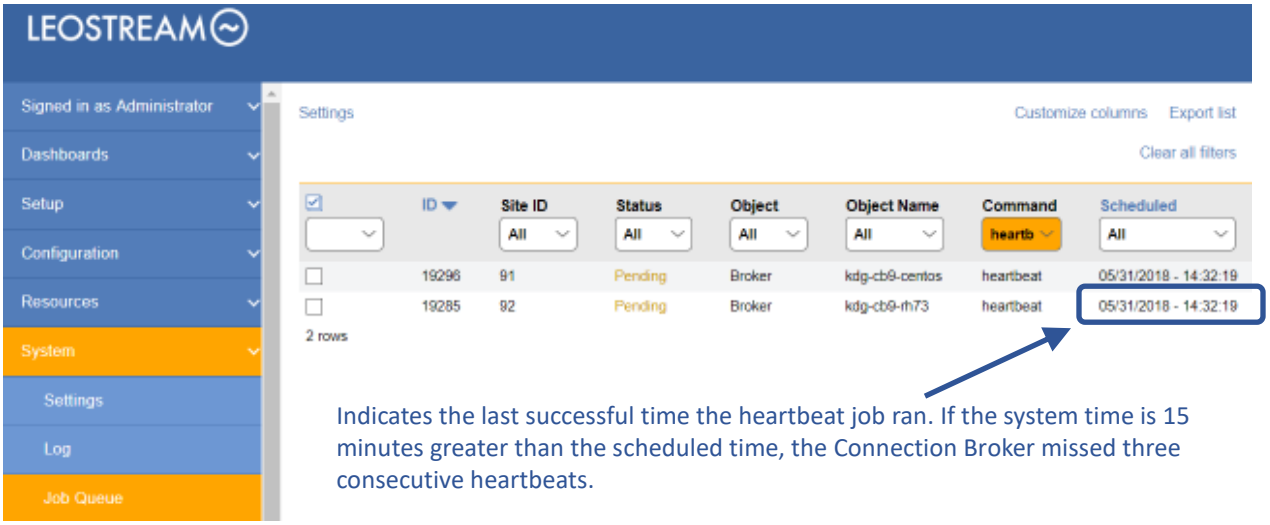
  The Connection Broker cancels the `heartbeat` job for a particular Connection Broker if the broker is powered off using options available on the **> System > Maintenance** page or from the virtual appliance console.

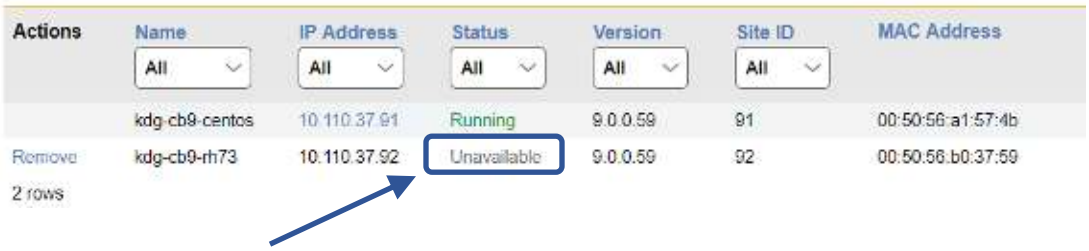
  When a stopped Connection Broker is powered back up, a new `heartbeat` job is added to the job queue, and the Connection Broker status updates to `Running`.

   The Connection Broker status is not properly updated if you power down the virtual appliance using power controls available in a virtualization management tool, such as vCenter Server. If you power down the virtual machine in any way other than through the VM console or using the **> System > Maintenance** page, you must wait for three consecutive heartbeat jobs to fail before the Connection Broker status is updated.

- **Unavailable:** Indicates that the cluster cannot determine the status of this Connection Broker. Unavailable Connection Brokers cannot process jobs in the job queue. The **> System**

> Cluster Management page marks a Connection Broker as unavailable after that Connection Broker misses three consecutive heartbeats. A missed heartbeat occurs when the `heartbeat` job associated with that Connection Broker cannot run. Because the heartbeat job attempts to run every five minutes, the Connection Broker is marked as unavailable after 15 minutes, as described in the following figures.





A Connection Broker can become unavailable due to connectivity issues or when it was powered off using the power controls in the virtualization environment in which the Connection Broker is installed.

Version

The Connection Broker version.

Site ID

The identification number used to represent each Connection Broker in the queue. Use the Site ID to determine which Connection Broker processed each job in the **> System > Job Queue** page.

UUID

The unique identifier for each Connection Broker.

MAC

The Connection Broker MAC address.

Booted

The day and time when the Connection Broker was last booted up.

Removing Connection Brokers from a Cluster

When building and testing your production environment, you may connect and disconnect any number of Connection Brokers from the external database at the cluster's core. Switch the Connection Broker back to its internal database, using the **Switch to internal database** option on the **> System > Maintenance** page, to remove the Connection Broker from the cluster.

When you remove a Connection Broker from a cluster all Finished, Cancelled, or Aborted jobs listed on the **> System > Job Queue** page are removed. Pending jobs remain assigned to the Broker.

 The Connection Broker cannot be removed from the cluster until it fails three consecutive heartbeat checks. Powering down a Connection Broker does not automatically remove that Connection Broker from the cluster.

To remove the Connection Broker from the cluster, after three heartbeat jobs fail and the Connection Broker status changes to **Stopped** or **Unavailable** on the **> System > Cluster Management** page, go to the **> System > Cluster Management** page and click the **Remove** link associated with the **Stopped** or **Unavailable** Connection Broker. When the Connection Broker is removed from the cluster, all pending jobs in the Job Queue are reassigned to other available Connection Brokers in the cluster.

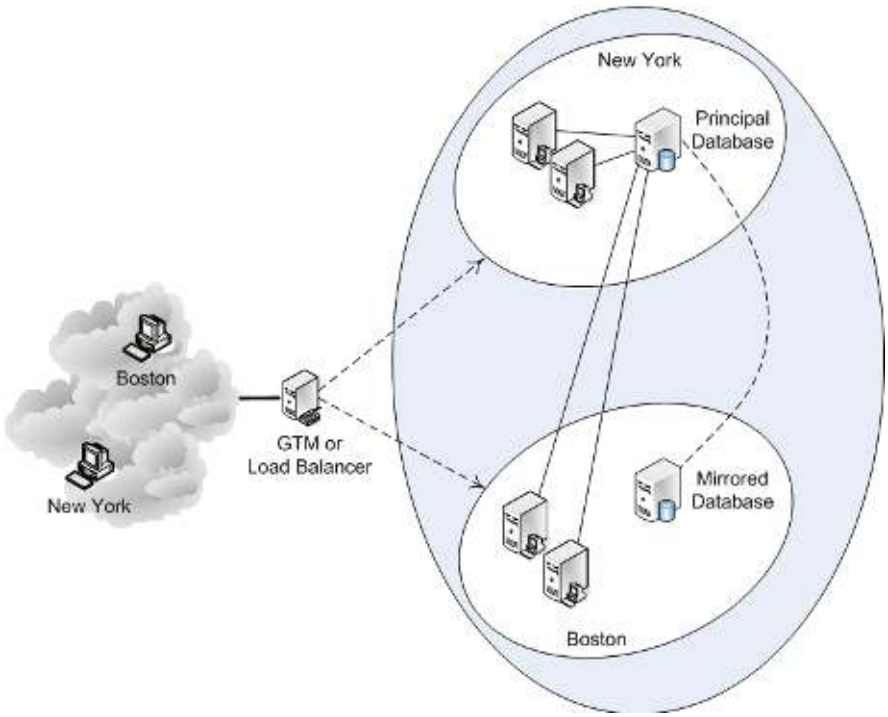
 The Connection Broker automatically rejoins the cluster and begins processing new Job Queue entries after it is rejoined to the cluster.

Updating Connection Broker Clusters to New Versions

All Connection Brokers in your cluster must run the same Connection Broker version. See the "Updating the Connection Broker" section in the Leostream Connection Broker Application Guide for instructions on how to upgrade the Connection Brokers in your cluster to the latest version

Spreading a Cluster across Multiple Datacenters/Regions

If your end users are spread across different regions, consider placing some of the Connection Brokers in your cluster in each region. By configuring your cluster to work with your global traffic management or load balancing systems, you can ensure that users log into the Connection Broker closest to their physical location. In addition, spreading your Connection Brokers across different regions provides disaster recovery and supports continued user logins in situations where a particular datacenter goes down.
Consider the following configuration.



This configuration consists of:

- A load balancer or global traffic management system
- Two Connection Brokers in New York
- Two Connection Brokers in Boston
- A principal SQL Server database in New York
- A mirrored database in Boston

All components, including components used solely as backups, should be continuously monitored to ensure that they are operational.

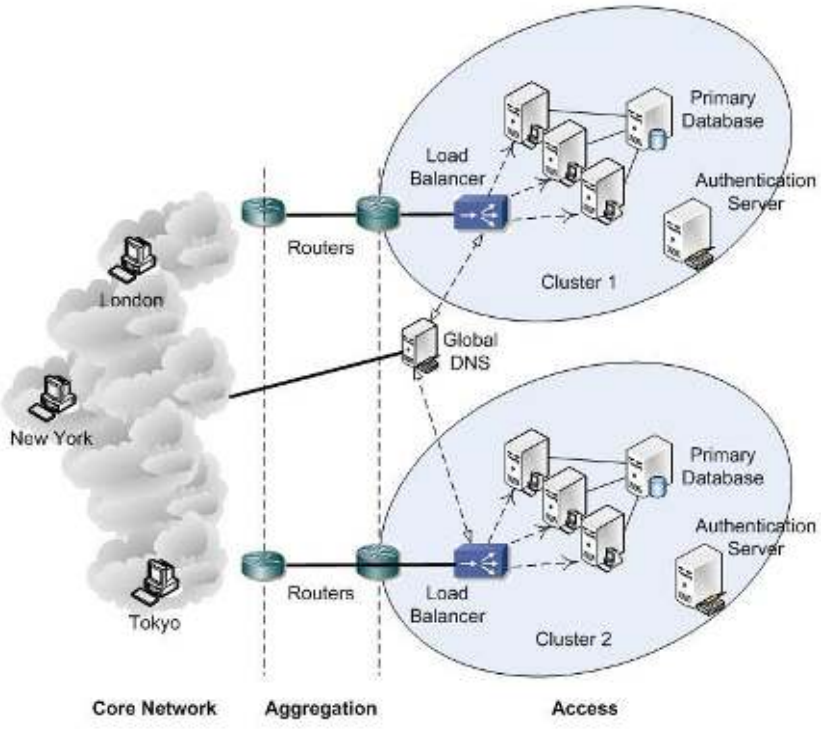
During normal operation, the New York Connection Brokers and Boston Connection Brokers connect to the principal Leostream database in New York. When a user logs in, the load balancer or GTM offers the Connection Broker closest to the user; New York users connect to a New York Connection Broker, Boston users connect to a Boston Connection Broker. Connection Brokers that are not processing user login jobs handle other work queue jobs. For example, if users are logging in only from New York, the Boston Connection Brokers process other (non-login) work queue jobs, reducing the load on the New York Connection Broker.

If the New York Connection Brokers stop responding, the load balancer directs New York and Boston users to the Boston Connection Brokers. If the New York primary Leostream database is still available, the Boston Connection Brokers continue to use that database. If the New York datacenter is completely unavailable and the principal Leostream database is offline, the mirrored database becomes the principal database (either manually or automatically, depending on the database configuration).

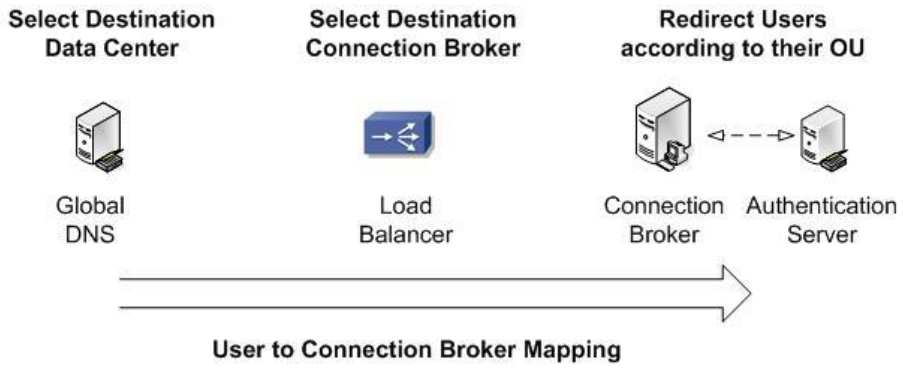
The Connection Broker times out the first database request after five seconds. After the first five second timeout, the Connection Broker makes two additional database connection attempts, each with a three-minute timeout. After a connection to the database is established, it is held open as long as possible. Although you could experience some five second timeouts over the WAN, the database connection should be made during the second or third timeout attempt.

Managing Different Clusters in Different Data Centers

If you want to handle a larger number of desktops or separately manage different region in your organization, you can create multiple clusters and use DNS to scale out across the clusters, as shown in the following figure.



There are three switch points that can be used to determine which Connection Broker a user logs into, as depicted by the following figure.



1. Global DNS determines the initial data center and cluster to use. Typically, a global DNS infrastructure redirects users to a particular data center according to a set of rules, often depending on the IP address of the user's client. This solution works well when there is a clear mapping between IP address ranges and locations, but does not work at all when a user moves to a different geographic location and requires access to their standard desktop. You can combine DNS with your Microsoft® Active Directory® service to use the domain membership of the client computer to determine the DNS response.

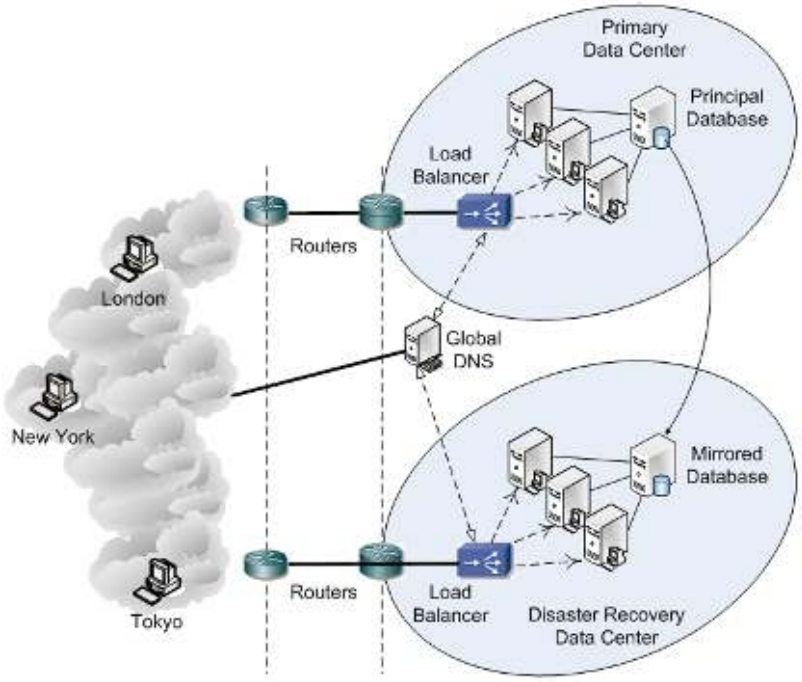
2. A local load balancer decides which Connection Broker in the cluster to use.

3. That Connection Broker, if necessary, redirects users to their home Connection Broker if they roamed outside their region and were incorrectly routed by Global DNS.

Individual clusters function independently of each other and, therefore, each cluster should manage a unique set of resources (virtual machines, blades, applications, etc.) although all clusters can manage the same users. If you do manage particular resources in multiple clusters, conflicts may arise. For example, in the figure at the beginning of this section, if Cluster 1 assigns desktop A to user A, Cluster 2 does not know about that assignment. Therefore, the Connection Brokers in Cluster 2 could offer desktop A to another user and, depending on Connection Broker settings, log user A out of their session.

Building a Cluster for Disaster Recovery

The section Spreading a Cluster across Multiple Datacenters/Regions shows how to support disaster recover scenarios using a single Connection Broker cluster. Instead, you can replicate your entire Connection Broker cluster in your disaster recover datacenter, as shown in the following figure.



In this case, the Global DNS directs the user to the primary or disaster recover datacenter, depending on the mode of operation.

Distributing User Logins and Network Traffic

For better Connection Broker availability, use a load balancer to spread user connections around the clustered Connection Brokers. All login traffic uses Web services, so your Connection Broker cluster behaves as a large Web server farm. If a Connection Broker fails, the load balancer redirects traffic away from the failed device.

Other Connection Broker features require different types of network traffic be routed appropriately to your cluster. For example, to receive disconnect notices from PCoIP Zero clients, syslog communications on port 514 must reach your cluster. For a complete list of ports used in the Leostream platform, see the Appendix A: Leostream Network Architecture.

There are a variety of algorithms for the load balancing calculation. Leostream recommends round-robin, with a *keepalive* for a particular Connection Broker based on load. The keepalive URL for a particular Connection Broker is:

```
https://CB_ADDRESS/index.pl?action=is_alive
```

Where *CB_ADDRESS* is your Connection Broker address. If the Connection Broker is processing a nominal load, the query responds with an HTTP status of 200 (OK) and displays CB_IS_OKAY in the Web browser. Once the Connection Broker becomes heavily loaded, the query returns an HTTP status of 503 (Service Unavailable). If the keepalive query returns status 503, route traffic away from that Connection Broker until the keepalive returns an HTTP status of 200 (OK) and displays CB_IS_OKAY in the Web browser.

The Connection Broker returns a status of 503 if any of the following conditions are met.

- The external database is offline.
- The Connection Broker load average is over 4. This is the first number in `cat /proc/loadavg`.
- Any of the active authentication servers defined on the **> Setup > Authentication Servers** page are offline.

 Because Connection Brokers run within virtual machines, their performance varies according to the overall load on that host, in addition to the load on the particular Connection Broker. Ensure that your Connection Brokers have sufficient resources on the host.

The load balancer also can use the Connection Broker XML RPC API to determine the health of the Connection Broker.

Citrix™ NetScaler™ Setup

Setup the Citrix™ NetScaler™ to perform two actions:

- Server monitoring
- Load balancing

 When configuring NetScaler for use with Leostream, ensure that you enable cookie persistence. For more information, review the following Citrix Knowledge Center article.

https://support.citrix.com/article/CTX205266

To monitor Connection Broker and database health use the HTTP-ECV functionality to allow the NetScaler to probe a particular URL on the Connection Broker. If it receives CB_IS_OKAY, the NetScaler application knows that the Connection Broker and the whole backend system are online.

Issue the following Web query to monitor the status of the Connection Broker:

```
https://CB_ADDRESS/index.pl?action=is_alive
```

Where *CB_ADDRESS* is your Connection Broker address.

If the Connection Broker is processing a nominal load, the query responds with an HTTP status of 200 `(OK)` and displays `CB_IS_OKAY` in the Web browser. Once the Connection Broker becomes heavily loaded, the query returns an HTTP status of 503 (`Service Unavailable`). If the keepalive query returns status 503, route traffic away from that Connection Broker until the keepalive returns an HTTP status of 200 `(OK)` and displays `CB_IS_OKAY` in the Web browser.

The following line gives the relevant command line for NetScaler.

```
> add monitor <name> http-ecv –send "GET /index.pl?action=is_alive" –recv
"CB_IS_OKAY"
```

For load balancing, use the Least Response Time, which is the time between the first request and the first byte of the first response that is returned.

- Use the `set lb vserver` command with an argument of `–lbmethod LEASTRESPONSETIME`.
- Set the persistence to 300 seconds.

F5® BIG-IP® Load Traffic Manager™ (LTM) Setup

Configure the F5® LTM system for both server monitoring and load balancing.

For server monitoring, use the Extended Content Verification (ECV) HTTP or HTTPS pre-configured monitors `http` or `https`. These monitors send a particular **Send String**, which must be set to `GET /index.pl?action=is_alive`, and expect to receive a particular **Receive String** of `CB_IS_OKAY`. Otherwise, the LTM system marks that Connection Broker as down.

Also, set the following parameters:

Load Balancing Method = Fastest Node
Persistence = Source Address Affinity

Connection Broker User Redirection

You can use the Connection Broker *user redirection* feature to redirect users to the appropriate Connection Broker and, hence, desktop. By redirecting the user, you can setup the user's policies and pools of desktops only in their home Connection Broker. If the user logs in through a different Connection Broker, that Connection Broker *forwards* the end user to their home Connection Broker.

 Your Leostream license determines if Connection Broker user redirection is enabled. If you require this feature, please contact sales@leostream.com.

Example: User Redirection Scenario

Consider the scenario where an American user goes to London, England, but needs to be connected to their standard desktop in New York. To accomplish this task:

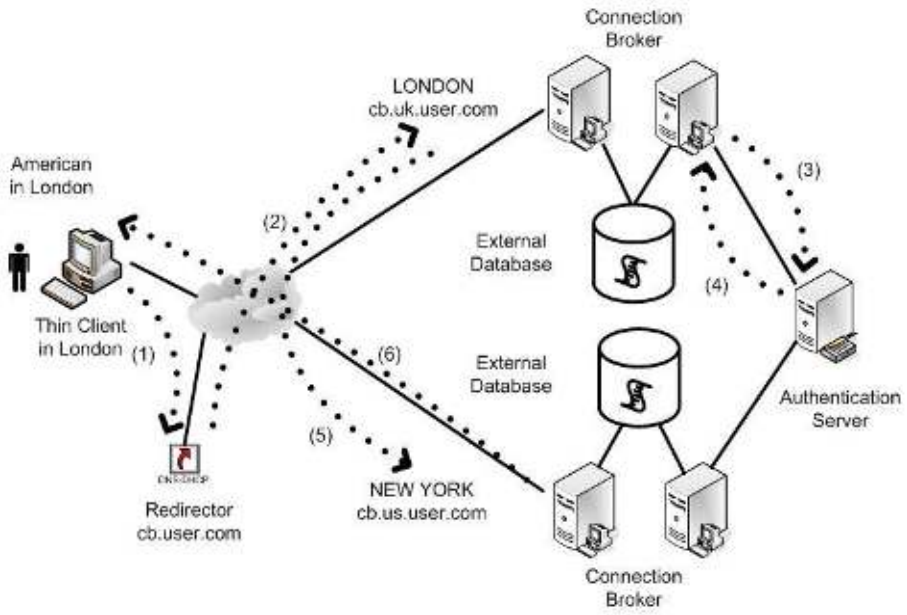
- Setup DNS in each region to point to the local Connection Broker cluster.
- Ensure that the authentication server used by each Connection Broker recognizes the American user.

- Configure the Connection Broker cluster in London to redirect the user to the New York Connection Broker cluster.

When the American user logs in through a thin client in London, the following occurs:

1. The thin client looks up `cb.company.com` in DNS and is sent to the Connection Broker at `cb.uk.company.com`.
2. The `cb.uk.company.com` Connection Broker looks up the user in the authentication system.
3. Based on the user's attributes in the authentication server, the `cb.uk.company.com` Connection Broker determines that the user should be forwarded to the New York Connection Broker.
4. The `cb.uk.company.com` Connection Broker sends a redirect command to the thin client, which includes the new DNS address for that user's Connection Broker, for example, `cb.us.user.com`.
5. The thin client receives the redirect and logs directly into the correct Connection Broker.
6. The `cb.us.user.com` Connection Broker instructs the thin client to connect to the correct desktop.

The following figure depicts the previous redirection scenario.



Setting up User Redirection in the Connection Broker

A Connection Broker determines a user's home Connection Broker using rules set up in your authentication servers. You can setup redirection, or *forwarding*, rules independently for each authentication server in the Connection Broker.

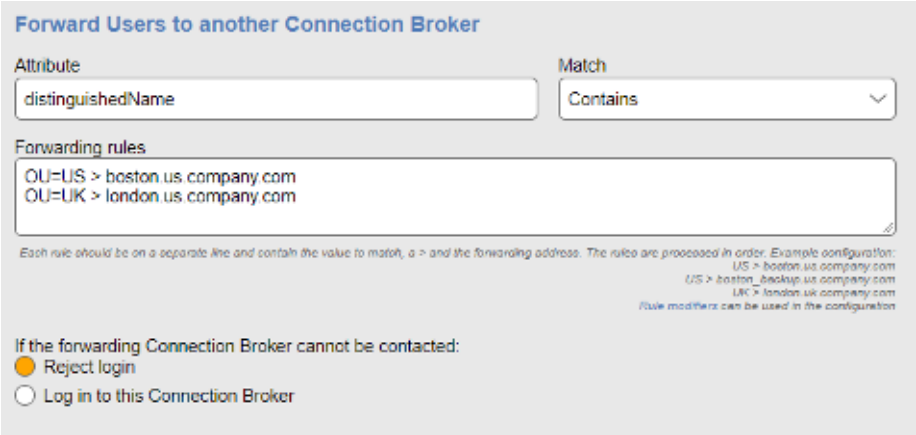
Set the redirection rules, as follows:

1. Go to the **> Setup > Authentication Servers > Edit** page for the relevant authentication server.
2. Scroll to the **Forward Users to another Connection Broker** section.
3. In the **Attribute** edit field, enter the attribute from within the authentication server schema to use to determine the criteria for redirection, for example, `distinguishedName`
4. Select a logic condition from the **Match** drop-down menu, which is used to compare the attribute to the list of possible values, for example, **Contains**.
5. In the **Forwarding rules** field, enter rules as a set of commands. Enter the list of attribute values and associated destinations as a series of rules in the following format:

   ```
   Value > Destination
   ```

   The Connection Broker tests the value on the left side of the greater than sign against the attribute entered in the **Attribute** edit field. If the string is found in the attribute in a way that satisfies the restriction in the **Match** drop-down menu, the Connection Broker returns the value on the right of the greater than sign.

   For example, in the following figure, if the user's `distinguishedName` contains the string `OU=US`, the Connection Broker forwards the user to `boston.us.company.com`:

   The forwarding rules are case sensitive.

   Enter one rule per line. The same value can be associated with different destination. In this way, when the Connection Broker searches down the list, if the first destination Connection Broker is not available, the destination associated with the next matching value is used.

6. Select the default behavior for when none of the forwarding rules apply:

   - **Reject login**: Does not allow the user to log in through any Connection Broker
   - **Log in to this Connection Broker**: Logs the user in through the local Connection Broker

Applying Forwarding only to Users that are already Assigned Desktops

User redirection can take into account whether a user has already been assigned a desktop by a particular Connection Broker cluster. To add this restriction, append the forwarding rule with the `if_assigned_only` flag, separated from the forwarding rules by one or more spaces. For example:

```
OU=US > boston.us.company.com if_assigned_only
```

In this case, the user is forwarded to the `boston.us.company.com` Connection Broker only if they are already assigned a desktop managed by that Connection Broker.

Using an External Database

In order to share information between Connection Brokers in a cluster, you must use an external data base. Leostream supports PostgreSQL version 9.1 or higher, Azure SQL, and Microsoft SQL Server when connecting to an external database. Leostream supports Microsoft SQL Server versions currently covered by Mainstream Support under the Microsoft Fixed Lifecycle Policy and versions in service under the Microsoft Modern Lifecycle Policy.


You cannot create a cluster that includes a mixture of version 8 and version 9 Connection Brokers. If you are upgrading to Leostream 9, detach all of your version 8 Connection Brokers from your external database before connecting your new version 9 brokers.

Sizing the External Database

Database Space Requirements

The Connection Broker uses the database to store all logs and information about each center, desktop, user, etc. Every desktop and user require approximately 1KB of storage space. Every user login and logout create approximately 5KB of log entry. By default, logs are retained for 30 days. Therefore, for example, if a user has five desktops that they access every day of the week, that user requires 150KB of database storage. As another example, a system with 1000 active users and 2000 desktops logging in once-a-day Monday through Friday requires approximately 150MB of database storage.


These estimates assume you have not deleted records from your system. For example, if you delete a center, the Connection Broker marks the desktop records associated with that center as deleted, however does not remove the records from the database. The database grows when you delete and recreate records. See Removing Deleted Database Records for information on when the Connection Broker purges records that are marked as deleted.

Database Transaction Requirements

Most of the load on the database occurs when users log into and log out of the system. When there is no user activity, the Connection Broker activity consists of tasks such as scanning centers, refreshing pools, checking Connection Broker heartbeats, etc.

While the load is split across multiple Connection Brokers, all brokers connect to a common database. Therefore, the load on the database rises with the number of logins per second. Each login request requires 30 database queries. A Connection Broker handling 5 logins a second generates 150 database queries a second. Three Connection Brokers handling 15 logins per second generates 450 queries a second.

To determine the hardware requirement, pick an industry benchmark. For this application, we use TPC-H (http://www.tpc.org/tpch/default.asp), an ad-hoc, decision support benchmark. Studying the TPC results suggests that a load of 75 logins per second can be comfortably handled by a four processor, with a total of eight cores 2.8 GHz processor system with 32G of memory.

Database Latency Considerations

The Connection Broker calls the database a number of times to query and configure information during user logins. Any latency in the connection between the Connection Broker and database server may slow down the login process.

In general, Leostream recommends you have less than 20ms of latency between your Connection Broker and database server.

Removing Deleted Database Records

When you delete a record from the Connection Broker, such as a user, policy, or center, the Connection Broker marks it (and any associated records, such as desktops from a center) as *deleted* in the Connection Broker database. Records that are marked as deleted are purged from the database after 90 days, plus the length of time the log is retained, as set by the **Days to retain log entries** option on the **Log Settings** page.

For example, if the **Days to retain log entries** option on the **Log Settings** page is set to 30 days, deleted log records are purged from the database after 120 days.

The following tables are exceptions to this rule. Items in these tables are purged, as follows.

- `log` entries are removed according to **Days to retain log entries** option on the **Log Settings** page
- `pool_history` entries are removed according to the selection in the **Retain data for** drop-down menu in the **Track historical pool assignments and connections** section on the **Edit Pool** page
- Deleted and completed records from the `work_queue` are removed after seven days
- `vc_host` entries are removed after two days
- Deleted `gateway_forward` entries are removed after two days
- Deleted `user_session` entries are removed after seven days
- Deleted `ad_attribute` entries are removed every four hours

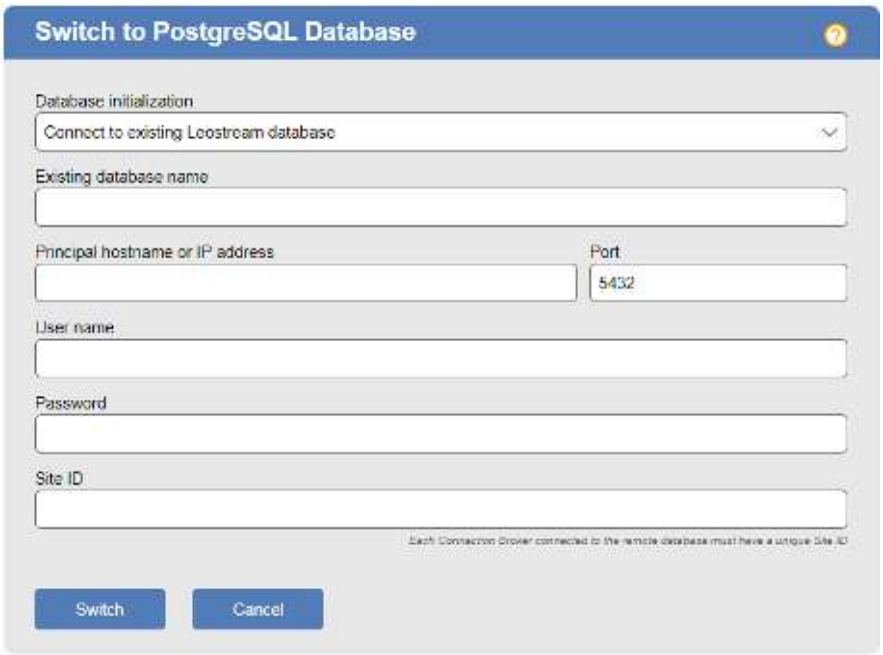
Switching to an External Database

The Connection Broker supplies an internal database that stores all configuration data when the broker is running as a standalone appliance. To enable Connection Broker clustering and failover, you must switch from the internal database to an external database. Leostream supports PostgreSQL version 9.1 or higher, Azure SQL, and Microsoft SQL Server 2014, 2016, or 2019 when connecting to an external database.

To switch to an external database:

1. Go to the **> System > Maintenance** page.
2. From the **Database** Options section, select the appropriate **Switch** option based on the type of database you plan to use and click **Next**.

   The **Switch database** form opens, as shown in the following figure when switching to a PostgreSQL database.



3. From the **Database initialization** drop-down menu, indicate if you are attaching to an existing database or if want to copy the contents of your current database to a new database.

   When connecting to an existing database that is populated with a Leostream configuration, the Connection Broker attaches to the database without copying any configuration information from its current database.
4. Enter the database name in the **Database name** edit field.
5. Enter the database server's hostname or IP address in the **Principal hostname or IP Address** edit field.

 You may create a DNS alias for your database server and use this DNS alias name as the hostname for the database.

6. Change the default outbound port listed in the **Port** edit field, if necessary.

    If you are using a named instance of Microsoft SQL Server, ensure that you enter the correct port number for that instance. You can view the ports associated with this instance in the **Protocols for instance name** dialog associated with this instance.

7. In the **User name** and **Password** edit fields, enter a username (including the domain, if applicable) and password for a user with access to the database.

   Under normal operation, the Connection Broker creates, deletes and updates rows in the database. During upgrades it may also create, delete and/or update tables and indices in the database. Ensure that you use a database user with the appropriate permissions, for example, for Microsoft SQL Server the user must have permission to support the following functions:

   - `db_ddladmin`
   - `db_datawriter`
   - `db_datareader`

8. Enter a unique **Site ID**. If you are using a cluster of Connection Brokers, each broker must have a unique Site ID.

   You can enter the site ID associated with a Connection Broker that was removed from the cluster. The new Connection Broker takes over any jobs in the work queue associated with the previous Connection Broker.

9. Click **Switch**. The Connection Broker takes one of the following actions:

The Connection Broker restarts after you switch databases.

If the Connection Broker loses its connection to the database, an error message appears in the Connection Broker logs. You can use that error message to issue an SNMP trap.

 You cannot currently connect to an external PostgreSQL database using SSL. Also, for newer versions of PostgreSQL, ensure that the `pg_hba.conf` file is configured to accept "password" authentication from remote connections.

For information specifically related to switching to PostgreSQL, Azure SQL, or Microsoft SQL Server, see Chapter 17 in the Connection Broker Administrator's Guide. For information on using the Connection Broker CLI to switch databases or change database parameters, see the Leostream Connection Broker Application Guide.

Database Failover

Specifying a Failover Database

After you attach your Connection Broker to an external PostgreSQL, Azure SQL, or Microsoft SQL Server database, you can specify a secondary database to use in the event the previously active database becomes unavailable.

If the Connection Broker is unable to contact the previously active database, the Connection Broker automatically switches to using the secondary database. At that point, the Connection Broker considers that to be the active database and continues to use that database until it becomes unavailable.

 You must ensure that your two databases remain in sync. Leostream does not replicate data between the databases. If a failover occurs, ensure that you properly replicate any changes made to the currently active Connection Broker database to the secondary database before bringing that database back online.

To specify a secondary database:

1. Go to the **> System > Maintenance** page.
2. Select the **Configure secondary database for failover** option.
3. Click **Next**.
4. Enter the name of the secondary database in the **Secondary database name** field. The database does not have to have the same name as your current external database, however ensure that the contents of the database matches that of your current database.

   Leostream does not perform any data validation when you save the form.
5. In the **Secondary hostname or IP address** edit field, enter the hostname or IP address of the database server that hosts the database.
6. Change the default outbound port listed in the **Port** edit field, if necessary.
7. In the **Secondary database user name** and **Secondary database password** edit fields, enter a username (including the domain, if applicable) and password for a user with access to the database. Leave these fields blank if the secondary database is accessible using the same credentials used for the current external database.
8. Click **Save**.

Using Microsoft SQL Server Always On Availability Groups

The Microsoft SQL Server Always On Availability Groups feature is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring.

A Microsoft SQL Server availability group supports a failover environment for a discrete set of user databases, known as availability databases, which fail over together. An availability group supports a set of primary databases and one to eight sets of corresponding secondary databases.

You can use the Always On Availability Group feature with your Leostream Connection Broker, to provide database failover for your Leostream environment. To set up your Connection Broker to use an availability database:

1. Create your SQL Server cluster and retrieve the cluster IP address from the **Cluster Core Resources** section of the **Failover Cluster Manager**.

2. Use the cluster IP when switching your Connection Broker to an external SQL Server database (see Switching to an External Database).

3. After you switch to the external database and the Connection Broker reboots, verify that the contents of the database correctly populated on all your SQL Server nodes.

Appendix A: Leostream Network Architecture