leostream

Remote Desktop Access Platform

Leostream® Platform – Scalability Guide

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

Version 202x
December 2024

Contacting Leostream

Leostream Corporation
77 Sleeper St.
PMB 02-123
Boston, MA 02210
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

Patents

Contents

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 availability, DR and capacity goals in a production-class environment, your site should 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.
- If using a hypervisor, 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.
- If using DevOps to deploy Connection Brokers, ensure the running version of the Connection Broker release is available for deployment
- 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.

High Availability Components

There are two workflows to consider when implementing for high availability:

1. **Leostream logins and desktop offers:** These are short transactions that consume few resources, but response time is visible to the end-user. High availability is important for these transactions; Load-balancing is not necessary. There are two types of situations to consider:

    a. Leostream Gateway availability for remote users

    b. Connection Broker availability for remote and internal users

2. **Desktop launches through the Leostream Gateway:** Remote desktop connections consume network bandwidth for the duration of the desktop session. These desktop sessions can benefit from high availability and load-balancing. There are two types of situations to consider:

    a. Remote users, who only use network bandwidth

    b. Leostream HTML5 users, who consume Leostream Gateway CPU in addition to network bandwidth

There are five components in a Leostream High Availability environment.

1. **Connection Broker:** The central Leostream platform component that processes business rules for managing user access to hosted resources
2. **Leostream Gateway:** The component that provides a network connection between the user and Connection Broker or the user and their remote desktop.
3. **Remote Desktops:** Computers located in the cloud or data center that are used by an individual or employee to perform work.
4. **Third-party database:** The component that stores information about the Connection Broker configuration settings and current state of the remote desktops
5. **Load Balancer:** A physical device or software application that distributes network traffic across multiple servers to improve the efficiency and reliability of applications. Some sites use DNS instead of a load balancer to distribute traffic. DNS does not monitor servers or applications but it can randomly direct traffic to different servers.

High Availability Architectures

High availability implementations should begin with a base foundation. Components are then added to increase the durability and resiliency of the environment. As more components are added to increase high availability, the complexity of the environment grows. Most sites do not require a very complex environment with 100% high availability. Instead, they choose the appropriate scenario for their company, based on their ability to withstand system outages, internal IT staffing and knowledge, and the company's IT resources and budget.

The following is an example of an implementation that starts with a base foundation. The Connection Broker concept of Locations is used to identify the area where the user logs in from, and their Leostream

Gateways, policies, desktop pools and protocol plans are configured to support the connection in the area.

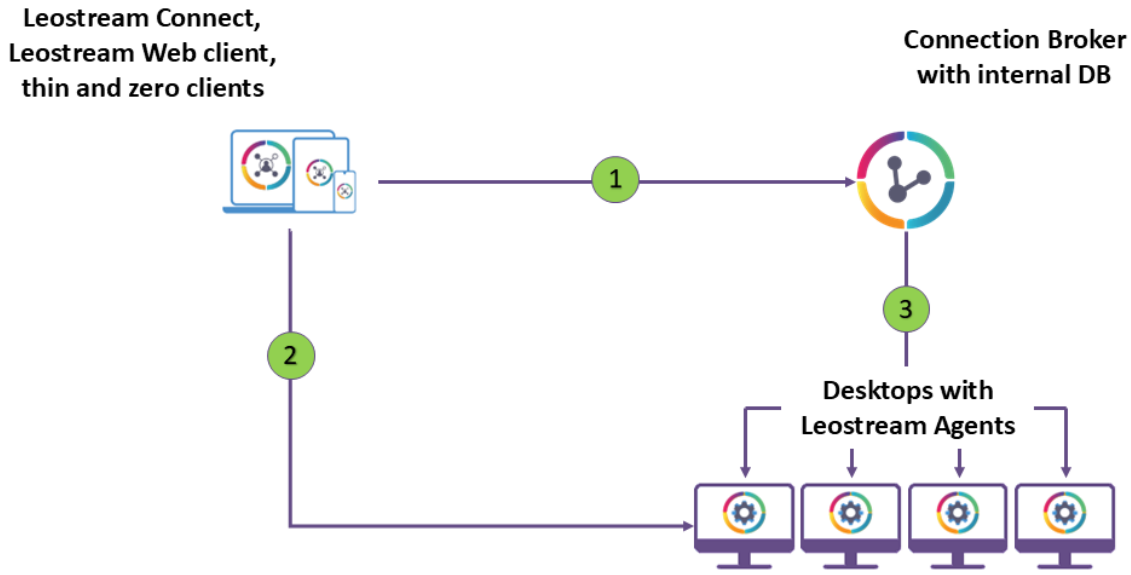
The scenarios that follow the example below continue to build on this base configuration to support different high-availability architectures. In each case, the numbers on the diagram correspond to the associated numbers for action in the scenario's description.

Single Connection Broker using an internal database

This implementation provides the foundation for a Leostream architecture. Use this architecture to test basic functionality, such as authentication, policy assignments, desktop offers and desktop connections. In addition, evaluation remote desktop performance using your chosen display protocol. This architecture does not provide any redundancy if a component fails so is suitable for a proof of concept, only.

In this scenario:

1. Users log in from their device to the Connection Broker on the internal network, and receive a desktop offer list.
2. Users select a desktop from their offer list. The desktop connection traffic travels from the client device directly to the desktop.
3. The Leostream Agents installed on the desktops communicate user activity such as logins, disconnects or logouts, to the Connection Broker.

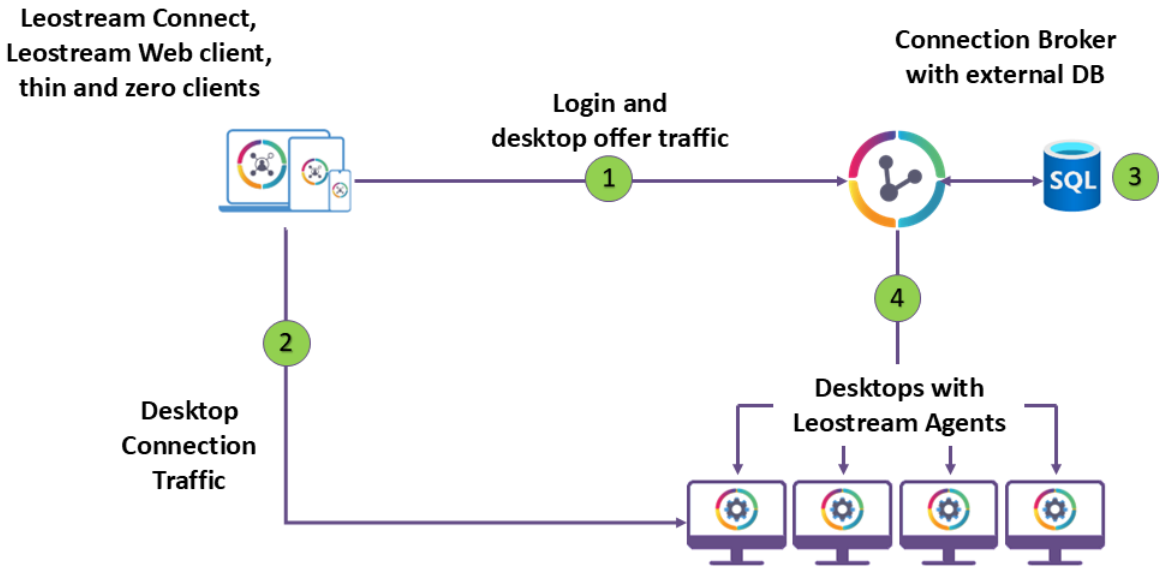


Single Connection Broker using an external database

This implementation introduces an external database in order to take advantage of your corporate policies and procedures used to maintain databases. The Connection Broker still uses its internal database to store the location of the external database, so backups should be taken in both environments.

In this scenario:

1. Users log in from their device to the Connection Broker on the internal network and receive a desktop offer list.
2. Users select a desktop from their offer list. The desktop connection traffic travels from the client device directly to the desktop.
3. Configuration, events, and logs are stored in the external database.
4. Leostream Agents on the desktops communicate user activity such as logins, disconnects or logouts, to the Connection Broker.



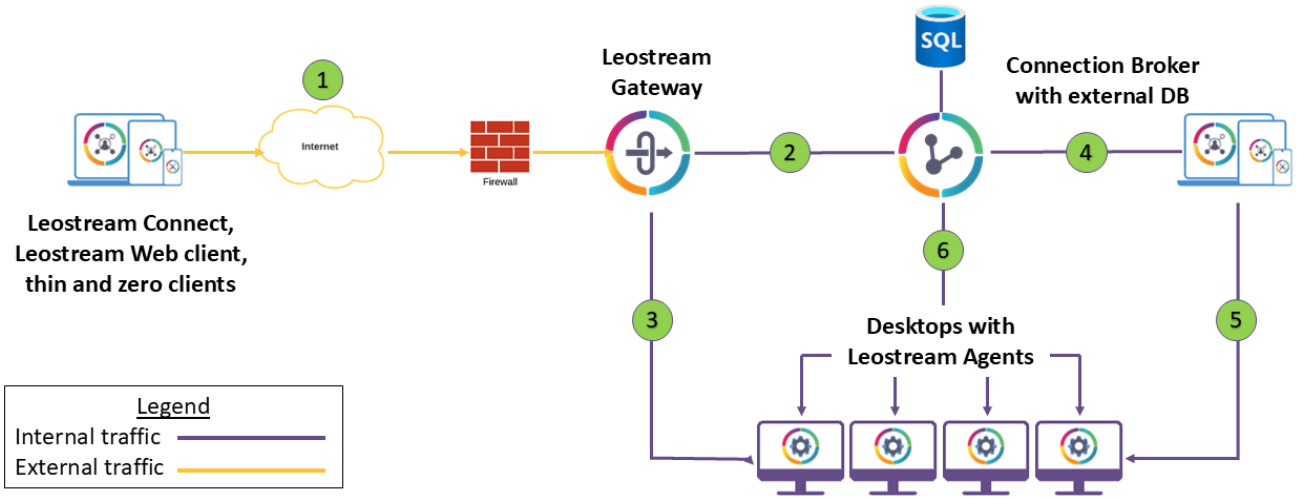
Single Leostream Gateway, forwarding to a single Connection Broker using an external database

This implementation adds remote users in a second location to the environment. Additional roles, policies and protocol plans may be configured to accommodate the remote workflow. Remote desktop performance should be re-evaluated, as the distance and number of network devices has grown between the user and their remote desktop.

This environment provides slightly more redundancy because users can work in a different environment if their internal environment is not physically available to them. It is suitable for a Proof-of-Concept or smaller environments that do not require high availability.

In this scenario:

1. Remote users log in from their device by entering the Leostream Gateway address.
2. The Leostream Gateway forwards login traffic to the Connection Broker on the internal network and users receive a desktop offer list.
3. Remote users select a desktop from their offer list. The desktop connection traffic travels from the client device to the Leostream Gateway, which redirects the connection traffic to the desktop.
4. Local users log in from their device to the Connection Broker on the internal network and receive a desktop offer list.
5. Local users select a desktop from their offer list. The desktop connection traffic travels on the internal network from the client device directly to the desktop.
6. Leostream Agents on the desktops communicate user activity such as logins, disconnects or logouts, to the Connection Broker.



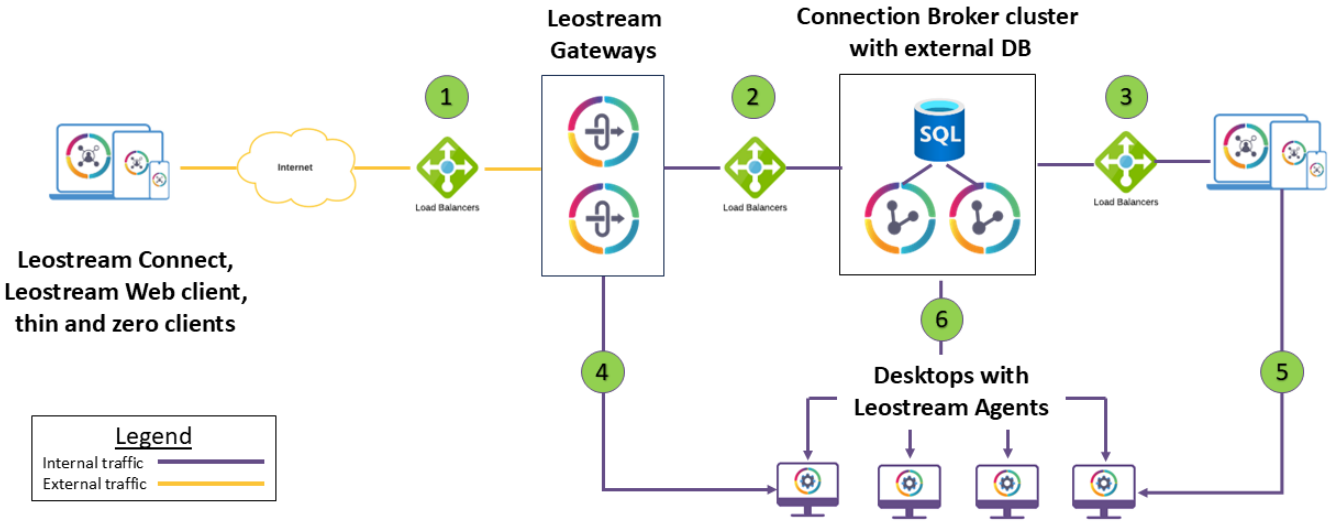
Multiple Leostream Gateways, forwarding to multiple Connection Brokers in a Connection Broker cluster, using an external database

This implementation introduces duplicate Leostream components to support redundancy. To achieve high availability, the Leostream Gateways should not reside on the same host or share the same disk storage. Likewise, the Connection Brokers should not reside on the same host or share the same disk storage.

The load balancer monitors the Leostream Gateway and Connection Broker environments and redirects traffic if a component is not responding. DNS can be substituted for the load balancer. Note that DNS can round-robin the network traffic, but it will not monitor the Leostream applications for performance or responsiveness. This implementation is suitable for smaller environments in one location.

In this scenario:

1. Remote users log in from their device by entering the load balancer address. The load balancer redirects the login traffic to one of the Leostream Gateways.
2. The Leostream Gateway forwards login traffic to the load balancer on the internal network. The load balancer, in turn, redirects the login traffic to one of the Connection Brokers. The Connection Broker returns a desktop offer list to the remote user.
3. Local users log in from their device by entering the load balancer address on the internal network. (This can be the same load balancer used in step 2, or a separate one as depicted in the diagram.) The load balancer redirects the login traffic to one of the Connection Brokers. The Connection Broker returns a desktop offer list to the local user.
4. The remote user selects a desktop from their offer list. The desktop connection traffic travels from the client device to one of the Leostream Gateways, which redirects the connection traffic to the desktop.
5. Local users select a desktop from their offer list. The desktop connection traffic travels on the internal network from the client device directly to the desktop.
6. Leostream Agents on the desktops communicate user activity such as logins, disconnects or logouts, to the Connection Broker.



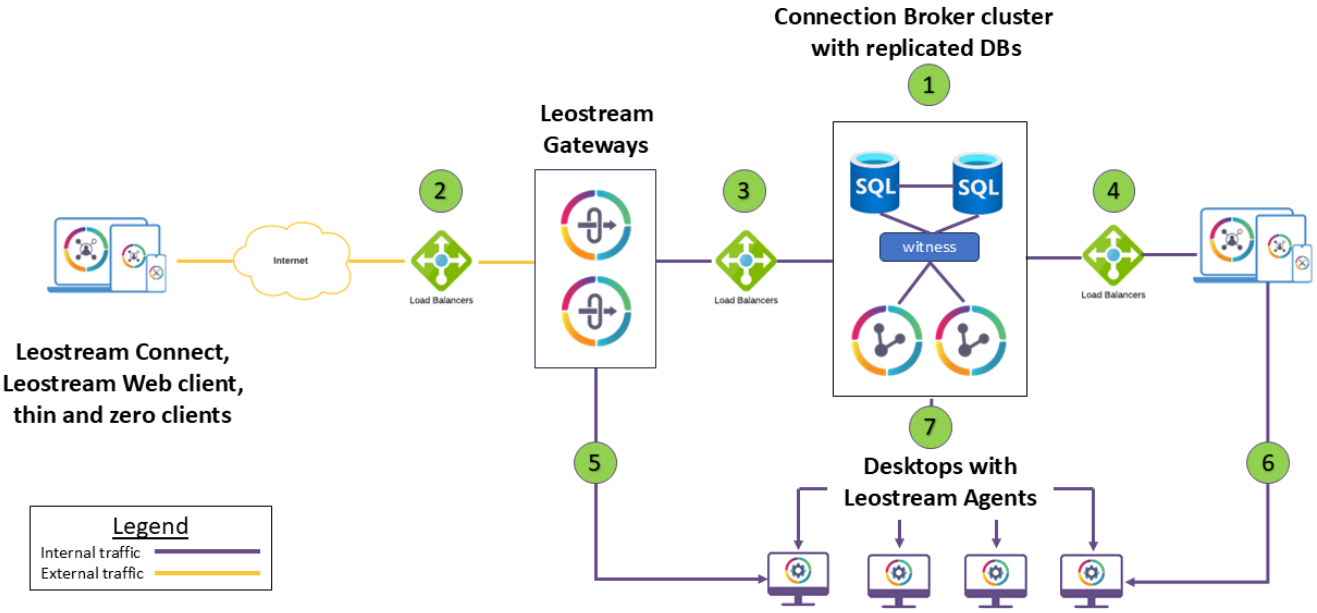
Multiple Leostream Gateways, forwarding to multiple Connection Brokers in a Cluster, using an external database with replication

This implementation adds database replication to reduce or eliminate downtime if the database server has issues. Replication does not affect Connection Broker performance. Database vendors provide different types of replications, using tables or logs or mirroring. Consult your DBA to determine the appropriate type of replication for your environment.

This implementation is suitable for environments that need 100% uptime.

In this scenario:

1. The Connection Broker cluster points to a database witness server. The database witness server redirects SQL requests to the primary database for processing. Data is replicated between the primary and backup database. The database witness server polls the primary database and promotes the backup database to primary if access to the original primary database is lost.
2. Remote users log in from their device by entering the load balancer address. The load balancer redirects the login traffic to one of the Leostream Gateways.
3. The Leostream Gateway forwards login traffic to the load balancer on the internal network. The load balancer, in turn, redirects the login traffic to one of the Connection Brokers in the cluster. The Connection Broker returns a desktop offer list to the remote user.
4. Local users log in from their device by entering the load balancer address on the internal network. This can be the same load balancer used in step 2, or a separate one as depicted in the diagram. The load balancer redirects the login traffic to one of the Connection Brokers in the cluster. The Connection Broker returns a desktop offer list to the local user.
5. The remote user selects a desktop from their offer list. The desktop connection traffic travels from the client device to one of the Leostream Gateways, which redirects the connection traffic to the desktop.
6. The local user selects a desktop from their offer list. The desktop connection traffic travels on the internal network from the client device directly to the desktop.
7. Leostream Agents on the desktops communicate user activity such as logins, disconnects or logouts, to the Connection Broker.



Multiple Leostream Gateways in different locations, forwarding to one Connection Brokers cluster, using an external database

This implementation introduces multiple remote locations as may exist in larger environments. The desktops for the users can reside in a different corporate data center or reside in the cloud. Alternatively, desktops can be in one cloud in different regions, or located in different cloud providers. Leveraging multiple locations ensures that the Leostream solution is isolated from site specific problems caused by IT resources or configuration changes.

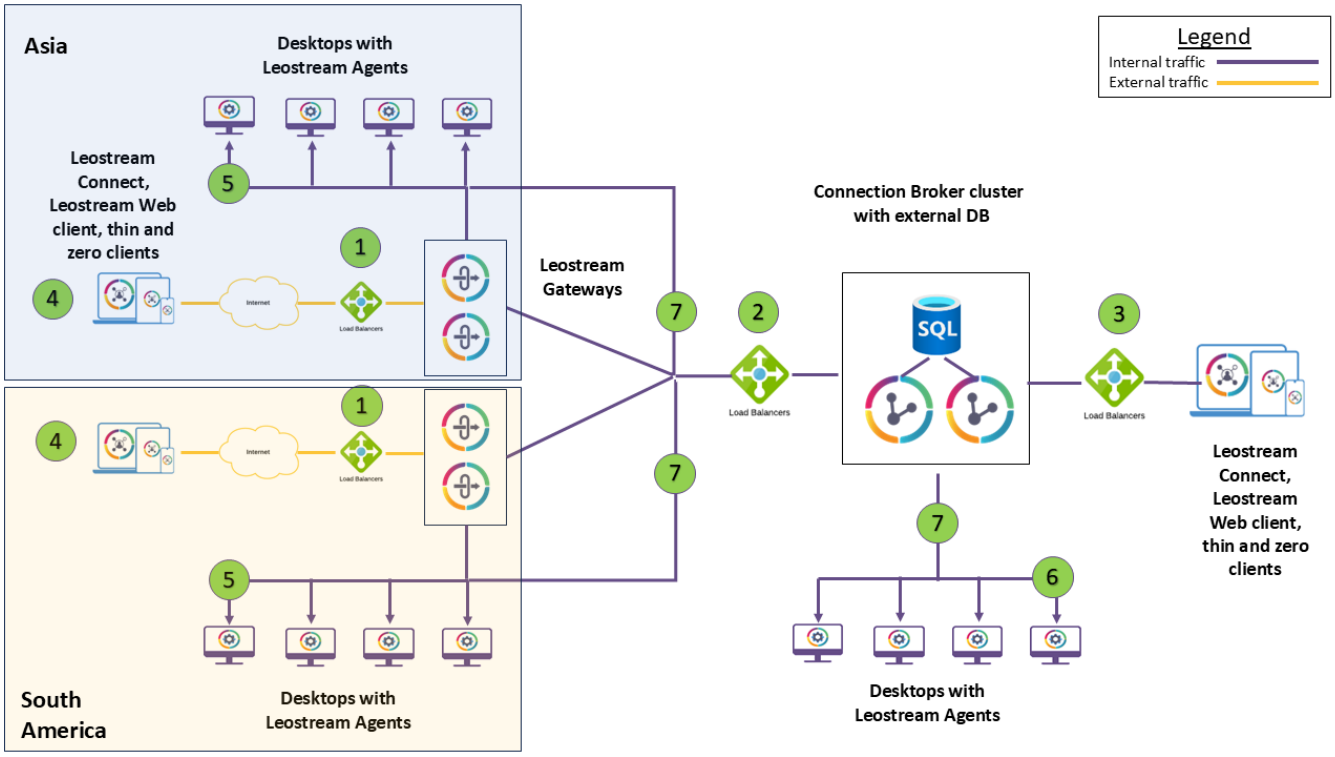
To achieve high availability and the best performance, the Leostream Gateways should reside in the same environment as the desktops. A single Connection Broker cluster can support the Leostream Gateways in different environments. The load balancer monitors the Leostream Gateway and Connection Broker environments and redirects traffic if a component is not responding. This implementation is suitable for complex, multi-site environments.

In this scenario:

1. Remote users log in from their device by entering the load balancer address in the region that is closest to them. The load balancer redirects the login traffic to a Leostream Gateway in that region.
2. Each Leostream Gateway forwards login traffic to the common load balancer on the internal network. The load balancer redirects the login traffic to one of the Connection Brokers in the cluster. The Connection Broker returns a desktop offer list to the remote user for desktops in their respective region.
3. Local users log in from their device by entering the load balancer address on the internal network. This can be the same load balancer used in step 2, or a separate one as depicted in the diagram. The load balancer redirects the login traffic to one of the Connection Brokers in the cluster. The

Connection Broker returns a desktop offer list to the local user.

4. The remote user selects a desktop from their offer list. Protocol plans are configured with the Leostream Gateway in the region closest to the user's desktop.
5. The desktop connection traffic travels from the client device to one of the Leostream Gateways in the region, which redirects the connection traffic to the desktop in same region.
6. Local users select a desktop from their offer list. The desktop connection traffic travels on the internal network from the client device directly to the desktop.
7. Leostream Agents on the desktops communicate user activity such as logins, disconnects or logouts, to the Connection Broker.



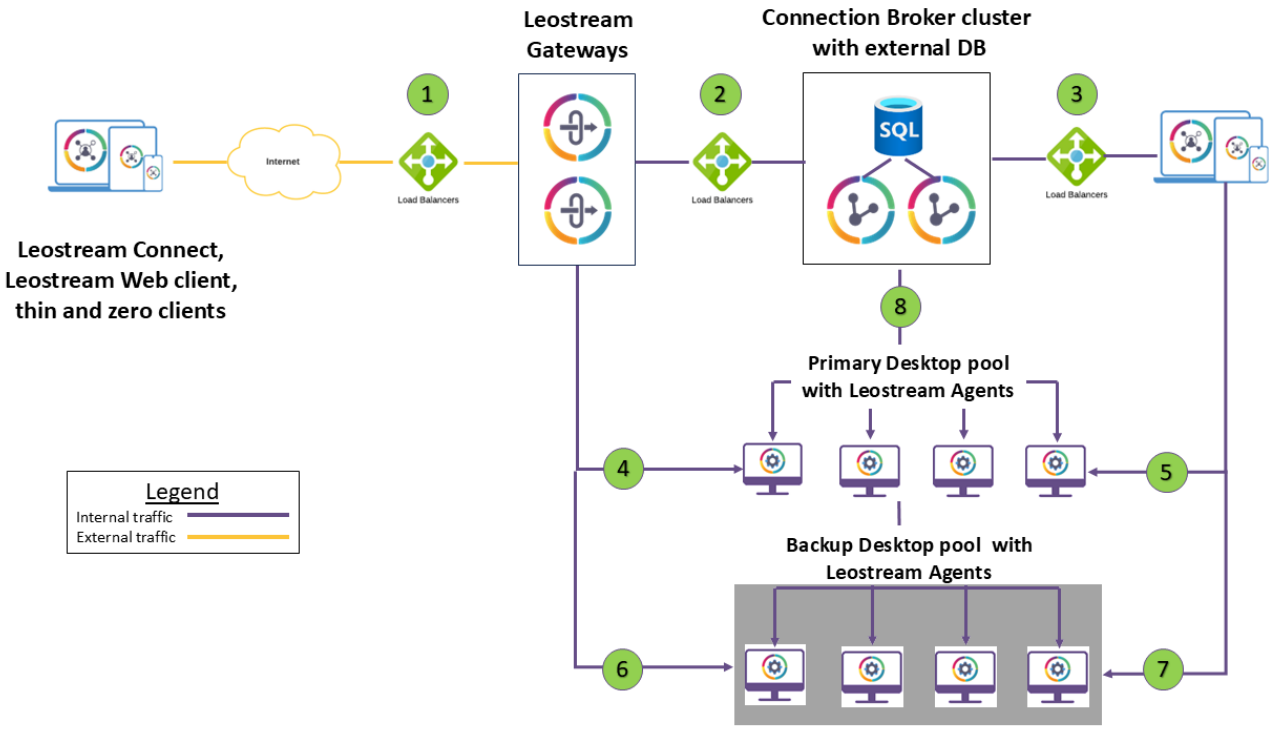
Leostream Gateways in a single location, forwarding to Connection Brokers cluster in a single location, using an external database and desktop backup pools

This implementation takes advantage of multiple desktop environments, which are available in some sites. The Leostream administrator determines the configuration of the primary and backup pools. The primary and backup desktop pools can reside in different locations or be different resources allocated from a single location.

Desktops are offered from the backup when the primary pool is exhausted. The load balancer monitors the Leostream Gateway and Connection Broker environments and redirects traffic if a component is not responding. This implementation is suitable for sites that have varying desktop resource demands or resource constraints within the primary pool environment.

In this scenario:

1. Remote users log in from their device by entering the load balancer address. The load balancer redirects the login traffic to one of the Leostream Gateways.
2. The Leostream Gateway forwards login traffic to the load balancer on the internal network. The load balancer redirects the login traffic to one of the Connection Brokers in the cluster. The Connection Broker returns a desktop offer list to the remote user.
3. Local users log in from their device by entering the load balancer address on the internal network. This can be the same load balancer used in step 2, or a separate one as depicted in the diagram. The load balancer redirects the login traffic to one of the Connection Brokers in the cluster. The Connection Broker returns a desktop offer list to the local user.
4. The remote user selects a desktop from their offer list, if a primary desktop is available. The desktop connection traffic travels from the client device to one of the Leostream Gateways, which redirects the connection traffic to the desktop in the primary pool.
5. The local user selects a desktop from their offer list, if a primary desktop is available. The desktop connection traffic travels on the internal network from the client device directly to the desktop in the primary pool.
6. The remote user selects a backup desktop from their offer list, if the primary pool is depleted. The desktop connection traffic travels from the client device to one of the Leostream Gateways, which redirects the connection traffic to the desktop in the backup pool.
7. The local user selects a backup desktop from their offer list, if the primary pool is depleted. The desktop connection traffic travels on the internal network from the client device directly to the desktop in the backup pool.
8. The Leostream Agents on the desktops communicate user activity such as logins, disconnects or logouts, to the Connection Broker.



High Availability Considerations

The previous implementation examples can be modified to fit your unique environment. Use the following guidelines when designing an environment that suits your site's needs.

The Leostream Gateway has three types of workloads that affect the number and placement of Leostream Gateways.

1. **Login forwarding:** The Leostream Gateway reroutes external login traffic to the Connection Broker. Other tools, such as firewalls or load balancers can provide this service if the network supports the connectivity. However, you may choose to use the Leostream Gateway to forward traffic, as it becomes a location identifier in the Connection Broker.
2. **Desktop connection forwarding:** Try to keep the remote desktops and Leostream Gateways used for the desktop connections networked as close as possible to the user's remote desktop. This will provide the most reliable and responsive desktop connection experience for the user.
3. **Leostream HTML5 display rendering:** The Leostream Gateways additionally can be used to display the desktop in a browser for the end user. As with desktop connection forwarding, you should try to keep the remote desktops and Leostream Gateway networked as close as possible. Additionally, the Leostream Gateway's CPU should be increased to support converting the desktop display into a web page to provide the most reliable and responsive desktop connection experience for the user.

The Connection Broker has three types of workloads that affect the number and placement of Connection Brokers.

1. **User login and desktop offers:** Users interact directly with the Connection Broker for login and desktop offers. These are quick transactions, which usually tolerate network distance. A single central Connection Broker cluster can support several Leostream Gateways distributed in different regions.
2. **Administrator configuration and monitoring**: Leostream administrators will appreciate quick access to the Connection Broker to monitor desktops and pool usage. When possible, the Connection Broker cluster should be in the same location as the administrators.
3. **Background job processing**: These jobs include monitoring components such as the Leostream Agents and Leostream Gateway, updating desktop inventory and processing user notifications. These jobs run in the background and do not affect user response time.

   The jobs are displayed on the Connection Broker **> System > Job Queue** page. The Connection Broker limits the number of background jobs that concurrently run. If jobs are routinely delayed because the limits are exceeded, add additional Connection Brokers servers to the Connection Broker cluster or increase the number of CPUs in the existing servers. See Appendix A for the job limits by Connection Broker server.

 The Connection Broker can tolerate database latency up to 20ms. Latency over 20ms can affect the end-user and administrator experience.

High Availability Checklist

When designing your Leostream environment, identify your expected uptime targets and user requirements before you start planning your high availability environment. Determining the most important goals sets expectations and helps to prioritize the Leostream implementation tasks. Some implementation tasks are dependent on others, such as switching to an external DB before deploying additional Connection Brokers for your Connection Broker cluster. Other steps are not dependent on others, but have less impact, so may have a lower priority.

The following is a sample task list for implementing High Availability:

1. Confirm access to remote desktops using the intended desktop protocols
2. Create a Leostream environment for testing on the internal network
3. Switch Connection Broker to an external DB
4. Establish a Connection Broker cluster using the external DB
5. Add Leostream Gateways for remote access
6. Add Leostream Gateway cluster for redundancy
7. Identify remote access using locations
8. Identify backup pools for remote desktop redundancy

9. Monitor environment to detect performance issues
10. Expand implementation to include additional Leostream Gateways, desktop pools, and locations for users in different locales.

 General Connection Broker configuration information is documented in the Connection Broker Quick Start Guide and Connection Broker Administrator's Guide. Configuration information for the Leostream Gateway is documented in the Leostream Gateway Guide . Additional detail regarding the steps for high availability are described in this document.

Using an External Database

To share information between Connection Brokers in a cluster, you must use an external database. Leostream supports PostgreSQL version 13 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.

 All Connection Brokers attached to the external database should run the same version of the Connection Broker. You cannot create a cluster that includes a mixture Connection Brokers versions.

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/), 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.

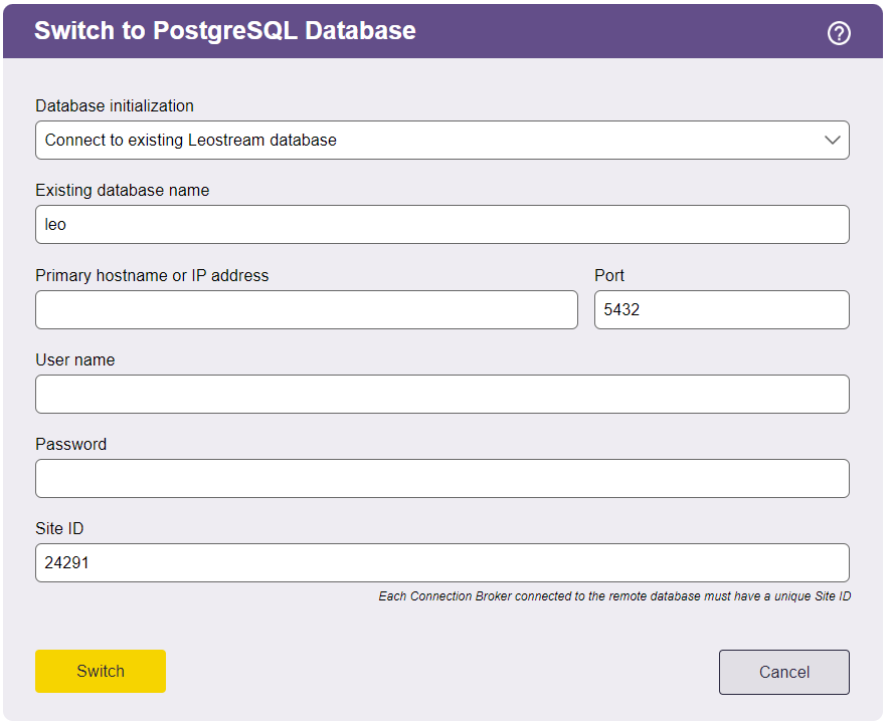
Switching to an External Database

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 13 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.

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.

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.

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
- `license_history` entries are removed according to **Days to retain log entries** option on the **Log Settings** page plus thirty days.
- `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

Using Connection Broker 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.

 All the Connection Brokers in a cluster must be on the same Connection Broker version.

 To support a large deployment and provide high availability, you can create a cluster of Leostream Connection Brokers managed by a third-party load balancer. The load balancer should be configured to distribute load and confirm that each Leostream Connection Broker is functional. See “Distributing User Logins and Network Traffic” chapter for additional information.

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 clusters use a common work queue, sharing jobs across all Connection Broker servers in the cluster.
- **Disaster Recovery:** Using clusters 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. 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.

  The number of concurrent background jobs running in the work queue is governed by the CPUs allocated to the Connection Broker server. The recommended 2 CPU minimum environment supports 12 concurrent jobs, as long as the load average is less than 3. New work queue jobs are not submitted until the load average is below 3.

  See Appendix A: Connection Broker Job Limits.

Creating a Cluster

To create a cluster of Connection Broker:

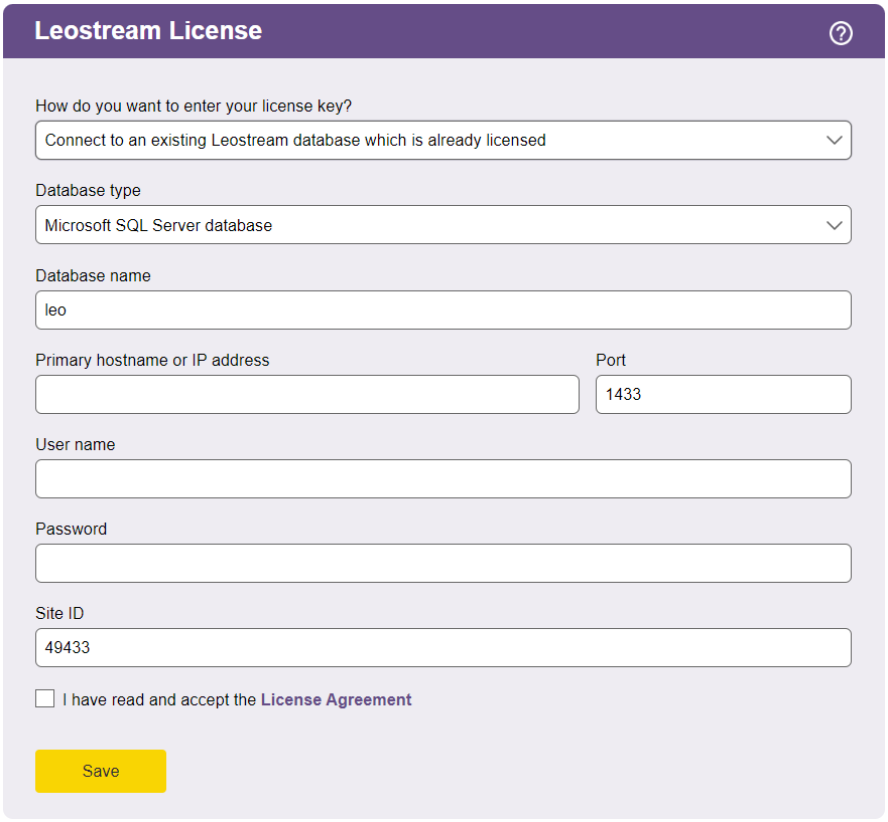
1. Install a standalone Connection Broker. By default, the Connection Broker uses an internal database.

    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 Connection Broker 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 attributes. You can modify the order and type of attributes 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 columns.

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 provided by the underlying operating system.

Status

Indicates the availability of each Connection Broker for processing requests, such as logins and running jobs in the job queue. Possible status values are as follows.

- **Running:** Indicates this Connection Broker is running and available to process requests.

- **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. If you decommissioned the Connection Broker, click the **Remove** action associated with that Connection Broker to remove its record from your Leostream database.

  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.

  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 are processed by the remaining Connection Broker servers in the cluster.

 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

Using Gateway Clusters

What is a Leostream Gateway Cluster?

To support a large deployment and provide high availability, you can create a cluster of Leostream Gateway. A Leostream Gateway *cluster* is a group of Leostream Gateways that support end-user access to the remote desktop. Each Leostream Gateway can operate individually and/or in a Leostream Gateway Cluster. The user's assigned protocol plan for the desktop connection determines if a Leostream Gateway or the Leostream Gateway cluster is used.

The Leostream Gateway also forwards login traffic to the Connection Broker, however Leostream Gateway Clusters are not used to forward login requests. Access to multiple Leostream Gateways for login traffic should be managed by a third-party load balancer or DNS. If you are using a load balancer, it should be configured to distribute load, as well as confirm that the Leostream Gateway is functional.

Creating a Leostream Gateway Cluster

Build a Leostream Gateway Cluster in your Connection Broker, as follows.

1. Create multiple Leostream Gateways on the **> Setup > Gateways** page and confirm they are working. Refer to the "Integrating with the Connection Broker" section of the Leostream Gateway Guide for instructions
2. In the **Add Gateway Cluster** form, enter a display name for your cluster in the **Name** edit field.
3. Indicate which Leostream Gateways should be configured to forward the display protocol traffic.
    a. **All Gateways in this cluster** – In this case, the Connection Broker allows your load balancer to control which Leostream Gateway forwards the display protocol traffic. Because the Connection Broker cannot predict which Leostream Gateway that your load balancer will choose, the Connection Broker instructs all Leostream Gateways in the cluster to open an appropriate firewall rule.

    When this option is selected, enter your load balancers IP address or resolvable hostname in the **Public IP address or FQDN of the external load balancer** field.

    This address must be accessible by the end users' client devices, and is the address used in Protocol Plans for the desktop connection. If you already attached a Leostream Gateway with this address to your Connection Broker, you must first edit that gateway record on the **> Setup > Gateway** page and reset the **Address** field to its private address.

    b. **The login Gateway** – If you enabled Connection Broker forwarding on your Leostream Gateways *and* your load balancer enforces sticky sessions, you can select this option to have the Connection Broker configure port forwarding only in the firewall of the Leostream Gateway that forwarded the user's login traffic.

    When using this option, the users protocol plan includes the address in the **Public IP address or FQDN for use in Protocol Plans** field of the Leostream Gateway that forwarded

the user login so this address must be available to the user's client.

This option allows more connections through your cluster, as Leostream isn't opening redundant ports on all the Leostream Gateways in the cluster.

c. **The Gateway with the fewest connections** – In this case, the Connection Broker selects the Leostream Gateway in the cluster with the fewest active desktop connections.

 After setting up your Gateway Cluster, ensure that it is selected in the **Gateway** drop-down menus in your Protocol Plan.

Dynamically configuring desktop access

The user's protocol plan determines if the desktop connection uses the Leostream Gateway to establish their desktop session. You may want to assign a protocol plan with a Leostream Gateway when a user works from home, but assign a protocol plan without the Leostream Gateway when the same user logs in from the office. Leostream uses Locations to identify when a user may need the protocol plan configuration with a Leostream Gateway, and when they do not.

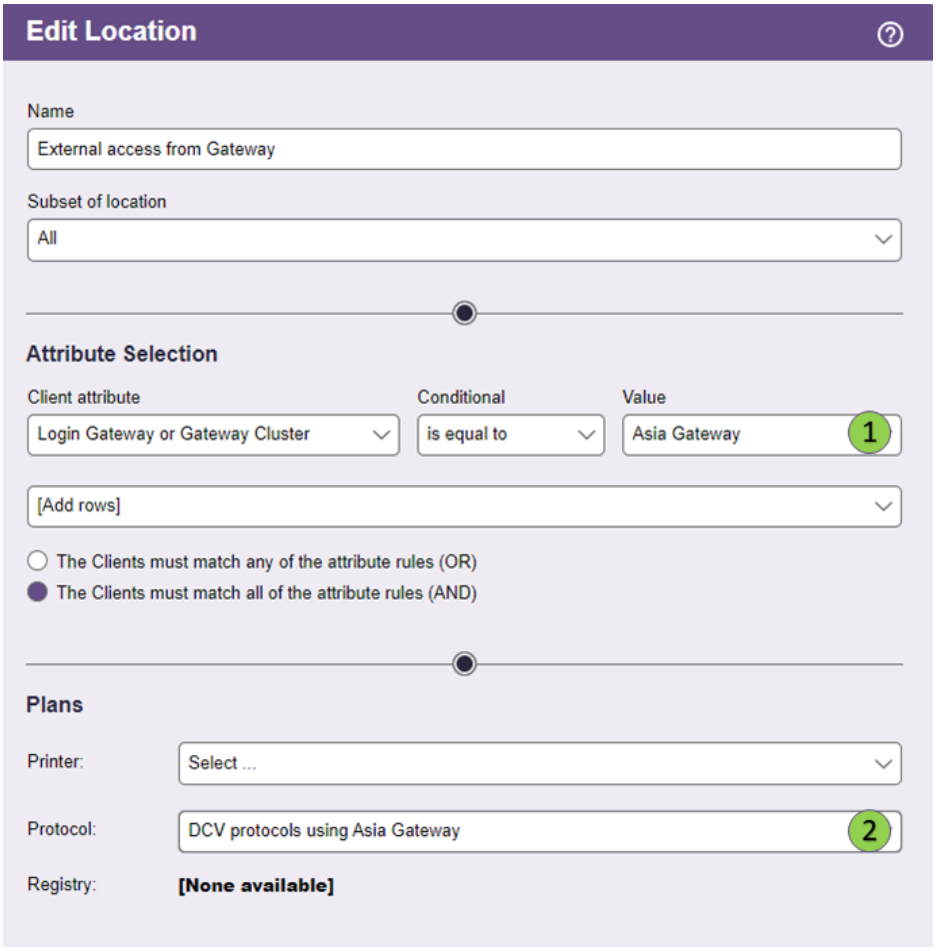
When a user logs into the Connection Broker from a client device, the Connection Broker registers the client device on the **> Resources > Clients** page. The Connection Broker also assigns that client to one or more locations. A *client location* is similar to a desktop pool, in that the location represents a group of clients with similar attributes. Locations can be created to identify clients that log in from physically different areas, using client attributes such as the Login Leostream Gateway or Gateway Cluster, IP masking and/or HTTP headers.

Locations can be used to assign the user's policy or to override the protocol plan assigned by their policy. When overriding the protocol plan based on location, the user is assigned the same policy and pool assignments on login, regardless of location. On desktop connection, then, the protocol plan corresponding to their location is used instead of the protocol plan in the policy.

Creating a Location

To create a location, navigate to the **> Configuration > Locations** page. The following figure shows an example location that includes all client devices that are forwarded to the Connection Broker by the Leostream Gateway named Asia Gateway.

1. The **Client Attribute** is set to **Login Gateway or Gateway Cluster**. Clients become members of this location if they were forwarded to the Connection Broker for login by a Leostream Gateway with a name that equals Asia Gateway.
2. Members of this location use the **DCV protocols using Asia Gateway** protocol plan. This protocol plan includes the Asia Gateway referenced in step 1.



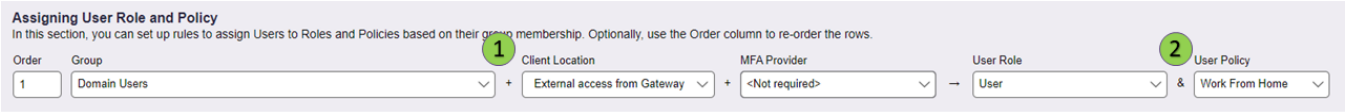
Gateway login traffic does not automatically determine the Leostream Gateway used for desktop connections. The Leostream Gateway used for desktop connections is explicitly defined in the protocol configuration section of the Protocol Plan. It is important to evaluate the login Leostream Gateway so the protocol plan with the correct Leostream Gateway is assigned.

Assigning a Policy Using a Location

User attributes along with the client's location decide which policy is assigned to the user. The policy determines the desktops offered to the user, along with the protocol plan assigned. The protocol plan contains the Leostream Gateway definition for desktop traffic.

To assign a policy using a location, navigate to the **> Configuration > Assignments** page. The following figure shows an assignment rule configure as described below.

1. User attributes and the user's client location is evaluated on login. The user must be a member of the Domain Users group and their client device must fall into the Externa access from Gateway location.
2. The Work From Home Policy is assigned that offers desktops and uses protocol plans appropriate for that location



Consider the location of remote desktops offered and Leostream Gateway used in the protocol plan when users log in in remotely.

Setting up Desktop Backup Pools

The Connection Broker determines the desktops offered to a user by evaluating the desktop pools in the policy assigned to the user at login. When a pool does not have available desktops, the user will either not be offered a desktop or receive a `No resources available` message. Alternatively, the Connection Broker allows you to specify a different, backup pool to ensure that the user receives a desktop in the event their primary desktop is unreachable. The desktops from the backup pool may be in a different physical location or cloud region, or may have different resource configurations. Backup pools and primary pools are created the same way. The only difference is how the pool is used in the policy.

Backup pools provide pool-based failover at *offer* time. In this case, when the user logs in, the Connection Broker selects a desktop from the primary pool and, at that point, determines if the desktop is reachable. If the desktop is not reachable, the Connection Broker selects a desktop from the backup pool.

Backup pools are available for hard-assigned desktops and policy-assigned desktops. When using backup pools, the user never sees which primary desktop they would have been offered and, therefore, do not necessarily know they are being connected to a backup desktop. Primary pool desktops are offered to the user if they are available when the user's offer list is refreshed, along with any currently assigned desktops from the backup pool.

See "Specifying Backup Pools" section in the Leostream Administrators guide for additional details

Distributing User Logins and Desktop Traffic

You may choose to control access to a Leostream Gateway or Connection Broker server by implementing a load balancer. A load balancer is a physical device or software application. It evaluates network traffic or application performance across a group of like servers. Using a load balancer can improve performance during peak times by distributing resources evenly across all servers in the load balanced group. Load balancers can also determine if a server is unresponsive and redirect traffic to other functional servers.

The Leostream Gateway and the Connection Broker integrate with most third-party load balancers. You should refer to the vendor's documentation for implementation information. You may also refer to the Leostream Support knowledge base for commonly used load balancer implementations.

The Connection Broker and Leostream Gateway are stateless applications that rely on APIs to improve scalability and portability. Therefore, it is important for the load balancer to maintain persistence for the Leostream applications. Load balancers used with Leostream must be configured to use **sticky sessions**.

Using Web Queries to Obtain Connection Broker Status

You need to setup application monitoring to effectively use a load balancer. You can monitor the Connection Broker using any of the following Web queries. These queries are useful, for example, if you use global or local load balancers and want to monitor the Connection Broker health at regular intervals.

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

Where *CB_ADDRESS* is your Connection Broker address.

These queries perform the following functions.

- `is_alive`: Responds with **`CB_IS_OKAY`** if all of the following conditions are true:

  1. The Connection Broker and its external database are online
  2. All authentication servers defined in the Connection Broker are available
  3. The Connection Broker load average is equal to or less than four

  Use the `is_alive` query with load balancers that direct user login requests. A Connection Broker that responds with **`CB_IS_OKAY`**, is ready to process the user login.

  If the Connection Broker cannot communicate with the database, the query returns an HTTP status of 503 (`Service Unavailable`). The query also returns an HTTP status of 503 (`Service Unavailable`) if the Connection Broker load average is above four or if any of the authentication servers defined in the Connection Broker are unavailable.

- `cb_status`: Responds with **`CB_IS_OKAY`** if the Connection Broker database is online. This function always returns a 200 Success header and returns an `ERROR_MESSAGE` if the database is not online.

  The `cb_status` query is lighter weight than the `is_alive` query and is a good option for performing general health checks on your Connection Broker. Leostream does not recommend using the `cb_status` query with load balancers that distribute user logins. A Connection Broker that responds to a `cb_status` query with **`CB_IS_OKAY`** may not be able to process user logins if, for example, an authentication server is offline.

- `cb_version`**:** Prints the current version of the Connection Broker when the Connection Broker application is running properly. Leostream recommends using the `cb_version` query in auto-scaling environments that are monitoring the Connection Broker application's health.

Autoscaling the Connection Broker Environment

Autoscaling is a feature available in cloud computing environments which dynamically adds or removes servers based on resource utilization. Sites may choose to implement autoscaling for the Connection Broker. The number of CPUs in the Connection Broker server and the corresponding load average should be considered when implementing your autoscaling environment. These metrics determine the maximum

number of concurrent Connection Broker jobs. The Connection Broker does not submit new background jobs if the number of running jobs exceeds the CPU limit or the server's 1 minute load average exceeds the value set for the server. Adding another Connection Broker server increases the number of background jobs that can run at the same time. Refer to Appendix A: Connection Broker Job Limits, for the maximum number of concurrent jobs in your environment.

Autoscaling may also deploy a new server when an existing server becomes unresponsive. Deploying a new Connection Broker server may not resolve the issue if the issue originates from an external system, such as the DB Server or Authentication Server. It is important to implement Connection Broker health checks before replacing a Connection Broker server in the environment.

⚠ Before your autoscaling system deletes or terminates a Connection Broker that fails a status call, check that your authentication servers and external database are healthy, communication from your Connection Broker to these systems is functioning properly, and all Connection Broker services are running. If the problem persists, please contact support@leostream.com prior to rebuilding your Connection Broker, as rebuilding the Connection Broker may destroy the records and logs required to diagnose the issue.

Checking the Leostream Gateway Status

If you use global or local load balancers, you should monitor the health of the Leostream Gateway at regular intervals. Use the following URL to check the status.

```
https://<your-gateway-address>/app/system/ping
```

The URL returns a status of `OK` if the gateway application is running.

Listing Leostream Gateway Connections

The Leostream Gateway should have enough network bandwidth to support the active desktop connections routed through it. Connection throughput varies based on the site's network and remote desktop workload. For example, a site with a handful of graphic-intensive applications may use more bandwidth than a site with many desktop connections with moderate network demands. Standard Linux commands, such as `nload`, `netstat` or `iftop`, can be used to monitor network traffic.

You should also monitor the number of Leostream Gateway connections associated with the network traffic. The `leostream-gateway` CLI displays the number of HTML5 connections currently hosted on a Leostream Gateway, using the `leostream-gateway --info` option.

```
sudo leostream-gateway --info
OS is CentOS/Rocky/Alma 8.6 - Rocky Linux 8.6 (Green Obsidian)
Port range is 20001-23000
Gateway version is 2024.2.1.2
Connection Broker forwarding is ON to cs-broker-demo
Azure Broker forwarding is OFF
Guacamole is ENABLED with 1 connection(s)
SELinux is DISABLED
The Gateway is RUNNING
This Gateway is attached to a Connection Broker
The Gateway signature is 9299a51bb89981d36c7b7f4fb0ce5585
The firewall zone is public
```

You can use the Leostream Gateway CLI to list all port-forwarded desktop connections, for example:

```
sudo leostream-gateway –conn
```

sudo leostream-gateway --conn

Desktop	Port	Dport	Source address	Key
---	---	---	---	---
10.112.112.230	20023	3389	10.254.251.104	336aa3d28cf54aa9f6a3521d258a5acd

The output lists the provides the following information:

- Desktop: IP address of the remote desktop
- Port: inbound port number used by the client device
- Dport: Port associated with the protocol on the remote desktop
- Source address: IP address of the client device.

 See the Leostream Gateway Guide for more information on the Leostream-gateway CLI.

Appendix A: Connection Broker Job Limits

The Connection Broker server runs background processes to support automation of the desktop environment. The number of background processes is limited by the number of CPUs and the corresponding load average for the number of CPUs. The following table describes the number of jobs allowed and the maximum load average associated with the CPUs. Background processes are delayed when either the maximum concurrent jobs or load average is met. The background processes resume when the number of jobs and load average are below the threshold.

CPUs	Maximum Concurrent Jobs	Maximum Load Average
2	12	3
3	16	4.1
4	20	5.2
5	24	6.3
6	28	7.4
7	32	8.5
8	36	9.6
9	40	10.7
10	44	11.8
12	52	14
16	68	18.4
32	132	36
64	260	71.2
128	516	141.6
256	1028	282.4

Appendix B: Leostream Network Architecture

Leostream relies on network access between the client, Gateway, Connection Broker and desktop. The following diagram provides the default network ports used between Leostream components and other components in the environment. A table representation of this information is also available in the Leostream Support KB article, Network Ports Used in a Leostream Environment