LEOSTREAM™ | OpenStack Guide

OpenStack Reference Architecture with the Leostream Platform

Advanced Capacity and Connection Management for your Hybrid Cloud

Version 9.0 – April 2018

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 direction, email sales@leostream.com.

Copyright

Trademarks

Patents

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

Introduction

For years, organizations around the world have deployed virtual desktop infrastructures (VDI) or offered desktops-as-a-service (DaaS) using full-stack, proprietary solutions. For many, however, the high cost and complexity of these solutions poses too large a barrier, and VDI adoption stalled at levels well below analyst predictions.

This reference architecture proposes an alternate solution, one that provides VDI to internal users as well as desktops-as-a-service to customers, at a lower cost and at Web scale.

Benefits of VDI and Desktops-as-a-Service

The reasons for moving from dedicated hardware to virtual desktops are well documented. Hosting desktops in the data center improves data security, allows organizations to save power, and centralizes desktop management.

These benefits are particularly important in verticals such as healthcare and financial services.
By keeping data secure in the data center, organizations are at a lower risk for losing data due to end-point loss or breach.

Desktops-as-a-service brings additional benefits to the SMB market. By relying on a managed service provider (MSP) to host their virtual desktops, SMBs can reap the benefits of VDI without making the costly investment into infrastructure and IT staff. Desktops-as-a-service are a benefit to the MSP, as well, allowing them to expand their portfolio, such as by providing disaster recovery (DR) solutions, leveraging the same data center used for their existing as-a-service offerings.

Why Use OpenStack® Software and Leostream for VDI and DaaS

OpenStack® software allows you to manage pools of compute, networking, and storage in your data center. When combined with the Leostream Connection Broker, your OpenStack compute becomes a VDI powerhouse, providing on-demand access to virtual desktops at lower cost.

An environment composed of OpenStack and Leostream enables some of the key aspects of VDI and DaaS, as described in the following sections.

Multi-tenant

Any hosted desktop solution used in a desktops-as-a-service environment must be multi-tenant. Not only do you need to manage tenants independently, each tenant's desktops must be isolated in their own network.

OpenStack projects provide multi-tenant management, as well as network isolation. You can separate customer instances and images by project, allowing you to easily track resource consumption for individual customers. Projects also allow you to set quotas, to ensure that particular customers do not overstep their allocated resource usage or negatively impact other customers.
OpenStack projects can define their own personal, virtual private cloud for each tenant, including IP address ranges, subnets, and routers. Only instances within a given private network, or those on subnets

connected through interfaces, can access other instances in that network, ensuring that individual customer desktops remain isolated from other customers.

The Leostream Connection Broker then provides multi-tenant management of users and desktop assignments. A single Leostream Connection Broker can authenticate users in different Active Directory domains, without establishing trust between the domains. Leostream centers, pools, and policies allow you to group OpenStack instances by projects and customers to ensure that users in different domains have access only to their allotted instances.

On-Demand Availability

A key aspect of a cloud environment is the fact that end users can request and quickly receive access to new, hosted resources. Using OpenStack with Leostream, you get on-demand availability for desktops.

OpenStack stores the base instances that you create using your preferred operating system and required applications. For each base instance, you create an image that is used by the Leostream Connection Broker to provision new instance as demand increases. New employees can be on-boarded in minutes by spinning up a pre-configured instance from one of your images.

This scenario also allows you to host legacy or one-time-use applications. For example, you can spin up a new desktop with the required application and terminate that instance when the user is done. Using this concept of a pool of preconfigured single-use desktops allows you to provide the user with the resource they need, without using up compute and storage resources when demand is low.

Less Expensive

Traditional single-stack VDI and DaaS solutions have proven to be costly to implement and license. Thankfully, OpenStack provides a viable alternative to these proprietary products.

Using OpenStack with the KVM hypervisor removes expensive virtualization licensing fees from the equation. In addition, the on-demand nature of OpenStack clouds means that you can provision and decommission desktops on a moment’s notice, optimizing the use of your storage and compute hardware.

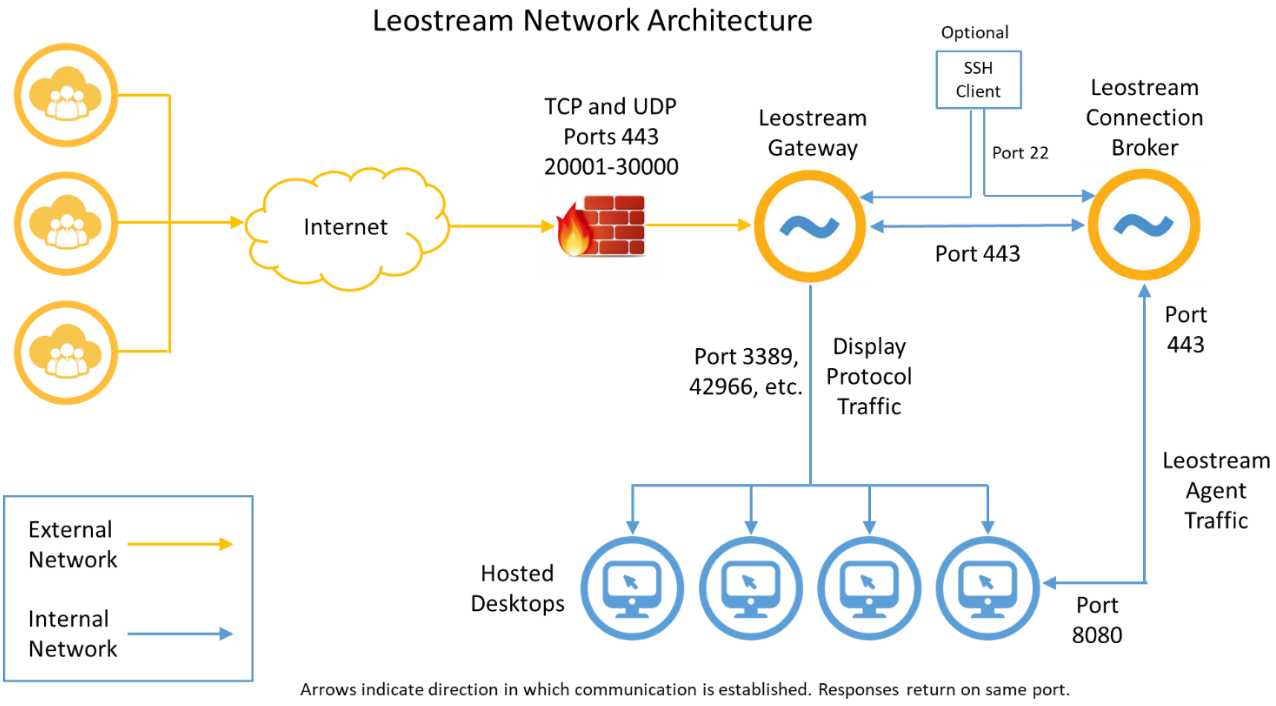
With OpenStack as the foundation, you can reduce the cost of deploying Windows desktops at scale, while gaining flexibility and benefits like desktop accessibility on any device. And, depending on your users’ needs, you can lower your costs even more by hosting Linux operating systems. The Leostream Connection Broker can manage VDI and DaaS environments that include both Windows and Linux operating systems.

Remote Access

The Leostream Gateway provides anywhere access to OpenStack instances that do not have a floating IP, allowing you to isolate instances for different customers or users. By integrating a Leostream Gateway into your environment, you can provide users with clientless HTML5-basesd access to both Windows and Linux machines. The Leostream Gateway also provides gateway functionality for high-performance protocols such as HP RGS and Mechdyne TGX, so you can satisfy even the pickiest user’s performance requirements.

High-Level Network Architecture

The following picture depicts a high-level network architecture of a Leostream environment.



Arrows indicate direction in which communication is established. Responses return on same port.

Reference Architecture Components

VDI or DaaS based on OpenStack includes the following components, described in the following sections.

- Infrastructure, including the hypervisor and storage
- OpenStack cloud operating system software
- Leostream platform
- Authentication servers, such as Microsoft Active Directory
- Display protocol

Infrastructure

OpenStack software runs on a wide range of hardware and in a number of configurations. A number of hardware vendors, including Dell, HP, and Cisco, provide detailed reference architectures that can help you determine what configuration is best for your needs.

This reference architecture does not include hardware recommendations. Please consult your preferred hardware vendor for information on the appropriate systems for running your OpenStack cloud.

Hypervisors

In the context of VDI, a key consideration is where you host the OpenStack compute instances, which are your users' desktops. These desktops can be hosted on any number of hypervisors or, using the OpenStack Ironic project, on bare metal systems.

OpenStack supports a number of hypervisors, including:

- QEMU / KVM
- VMware vSphere
- Microsoft Hyper-v
- Citrix XenServer
- Xen via libvert

Not all hypervisors provide the same level of functionality in an OpenStack environment. Consult the OpenStack documentation for the latest matrix on hypervisor support. Any of the supported hypervisors are sufficient for use in a VDI/DaaS environment. You can use the KVM hypervisor provided with most OpenStack distributions to avoid any additional licensing fees associated with vendor-specific hypervisors.

When using the OpenStack Ironic project for bare metal provisioning, you can investigate solutions such as HP Moonshot System. HP Moonshot Systems host desktops on individual SoCs (System-on-Chips), which can provide better performance for certain workloads.

Hosting desktops on bare metal may also open up new Microsoft operating system licensing models, as virtualizing Windows client operating systems require special hardware considerations. Please, consult your Microsoft licensing specialist for the most up-to-date information on using Microsoft Windows operating systems in an OpenStack environment.

Leostream can manage OpenStack VDI and DaaS using any hypervisor or physical system to host the compute service. You can design your infrastructure in the manner best suited to your licensing and capacity needs. In addition, Leostream supports a heterogeneous environment, so you can use a mixture of hypervisors and bare metal systems.

Storage

OpenStack includes a number of different storage methodologies.

- Root disk – Root disk storage is managed by Nova on a compute instance, and runs the operating system for the instance. The root disk persists between instance reboots, and is backed up when a snapshot is taken of the instance.

- Ephemeral – Ephemeral storage is additional storage managed by Nova that can be associated with an OpenStack instance. Ephemeral disks are similar to root disks in that the data is retained between instance reboots, however the disk is destroyed when the instance is terminated. Also, ephemeral storage is not backed up during an OpenStack instance snapshot.

- Block – Cinder block storage provides additional disks that can be used for storing user data. Cinder volumes can be detached from one instance, and reattached to another instance, providing persistent data across the lifespan of several instances.

- Object – Object storage is most useful when managing large datasets.

When working with desktop loads in VDI, root and block storage play the most important role. Ephemeral storage is not recommended as the data stored on the ephemeral disks are not backed up as part of an instance snapshot. Object storage works very well for unstructured data sets where data is generally read but not written-to, which, again, is not appropriate for VDI.

Consider two scenarios.

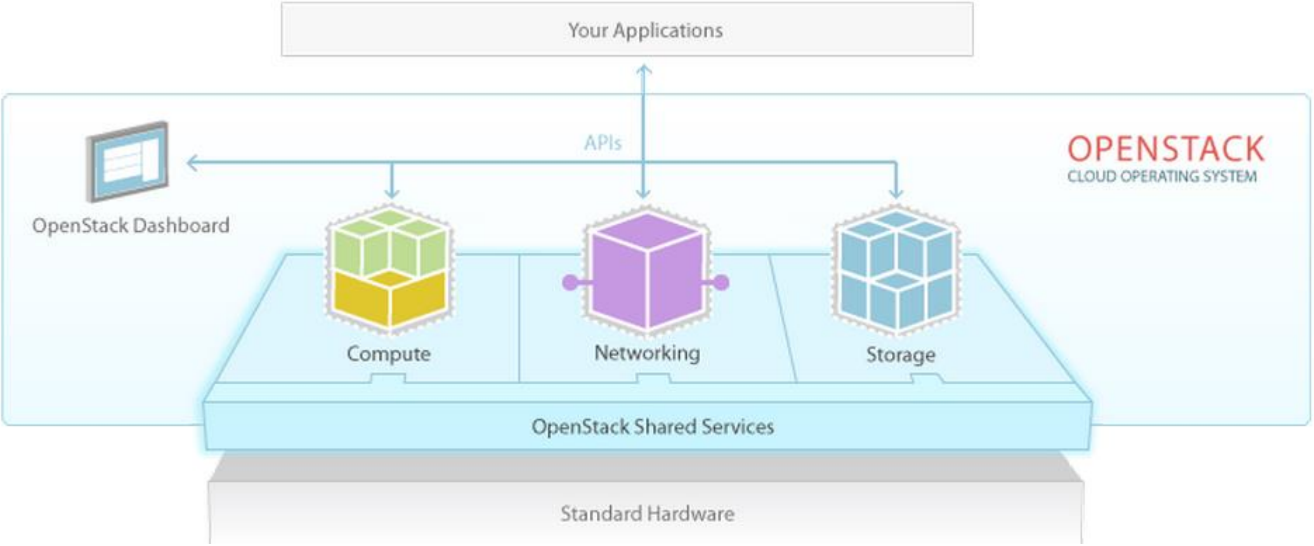
- **Permanent** instance – A permanent desktop is an OpenStack instance that is never terminated. Data stored on the root disk is retained between reboots, and for the entire lifetime of the desktop. A permanent desktop can be persistently assigned to a particular user, and their data can be stored on the root disk. Alternatively, a permanent instance can model a shared, non-persistent desktop if personal user data is stored off of the root disk, using Cinder block storage or any other file-sharing system.

- **Single-Use** instance – A single-use desktop is an OpenStack instance that is terminated as soon as the user logs out of the desktop. Personal user data must be kept off of the root disk, as it is deleted as soon as the instance is terminated. In this case, Cinder block storage or another file sharing system can be used to store user data off of the instance.

This reference architecture focuses on scenarios where user data is stored on the root disk. Cinder block storage is not covered as part of this documentation.

OpenStack

OpenStack software controls large pools of compute, storage, and networking resources throughout a data center. OpenStack is comprised of a number or projects, each focusing on a particular aspect of the cloud operating system. The following figure, taken from the **OpenStack website**, shows most of the projects that are important for hosting VDI workloads in OpenStack.



In particular, the following OpenStack projects are considered in this reference architecture. Other projects may be useful in a VDI environment, but are not covered by this documentation.

OpenStack Project	Service	Description
Nova	Compute	Provides scalable, on-demand compute resources, or virtual machines, for the VDI environment
Neutron	Networking	Provides on-demand, scalable, and technology-agnostic network abstraction
Keystone	Identity	Facilitates client authentication and authorization
Glance	Image	Manages images that can be used to spin up new compute instances
Cinder	Block storage	Provides block volume storage that can be used to store persistent user data
Horizon	Dashboard	Provides a Web interface that can be used to manage the OpenStack environment setup
Ironic	Bare metal	Manages and provisions OpenStack instances onto physical machines

This reference architecture does not cover designing your OpenStack environment. Please, consult your OpenStack experts for information on building a resilient, scalable, fault-tolerant cloud environment.

Leostream Platform

Leostream lies at the heart of any hosted resource deployment, providing crucial functionality for assigning desktops to users and managing their connections. For cloud environments, Leostream also provides advanced functionality for managing capacity, allowing you to expand and contract your cloud environment to meet the ever-changing demands of your organization.

For more information about Leostream Connection Broker concepts, see the **Getting Started with Connection Broker Concepts** guide.

Leostream Components

The Leostream environment consists of the following four components.

- **Connection Broker**: The main application that manages the hosted desktop environment. The Connection Broker is the central management layer for configuring your deployment, including inventorying and provisioning desktops, assigning and connecting users to these desktops, and defining the end-user experience. The Connection Broker also includes a web portal for users to access their hosted resources.
- **Leostream Gateway:** An optional application that provides HTML5-based clientless remote access for users connecting to their remote desktop. The Leostream Gateway also provides gateway functionality for protocols such as RDP, HP RGS, and Mechdyne TGX, to connect users to desktops

that are hosted in a network that is isolated from the user's client device.

- **Leostream Agent**: When installed on the remote desktop, the Leostream Agent provides the Connection Broker with insight into the connection status of remote users, including when they log out, disconnect, or lock their desktop. The Agent also manages enhancements such as USB device passthrough and network printer redirection. The Leostream Agent is available for Microsoft Windows, Linux and macOS operating systems.

- **Leostream Connect:** A software client provided by Leostream that allows users to log into your Leostream environment and access their hosted resources from fat or thin clients. Using Leostream Connect, you can repurpose existing desktops and laptops as client devices, lowering the cost of VDI deployments. Some thin clients provide built-in Leostream Connect clients.

  In addition to using Leostream Connect, users can log into Leostream using the Leostream Web client, any PCoIP client device, Dell Wyse ThinOS clients, or any number of compatible thin clients.

- **Database:** In a proof-of-concept environment, the Connection Broker stores all information in an internal PostgreSQL database. A large-scale, redundant production environment requires an external PostgreSQL or Microsoft® SQL Server® 2012, 2014, or 2016 database.

Authentication Servers

Authentication servers, such as Microsoft Active Directory, are responsible for authenticating users into your VDI or DaaS environment. The Leostream Connection Broker can act as a local authentication server, if you do not need domain users.

In a VDI or DaaS environment that includes Leostream, the user's record in your authentication server determines which Leostream policy the user is offered and, therefore, which OpenStack instances they may access. You may include any number of authentication servers in your Leostream environment, without establishing any trust relationships between the domains. For DaaS environments, this ability allows you to manage multiple customer accounts in a single Connection Broker.

Leostream can authenticate users against the following authentication servers.

- Microsoft Active Directory
- OpenLDAP
- NIS

Display Protocols

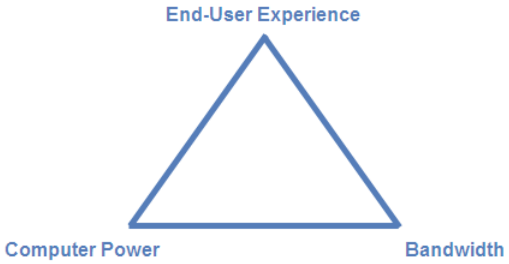
The display protocol is responsible for remoting the graphical information from the remote desktop to the user's client device. Leostream can establish a connection to a remote desktop using a variety of supported display protocols.

After the connection is established, the Connection Broker removes itself from the connection path, i.e., the Connection Broker does not proxy the remote session. This is important to note in an OpenStack environment that isolates instances on a private network. You must provide access into the private network using a floating IP address, VPN, or other solution.

Choosing the right protocol requires a balance between the need for a good end-user experience, the bandwidth available on the network, and the compute power supplied by the hardware. Every display protocol struggles with the task of satisfying these requirements, with the ultimate goal being:

- Low bandwidth
- Low computational requirements
- High-quality end-user experience

These three factors make up the *protocol triangle*, depicted in the following figure. As with any triangle, changing the angle for one corner always has repercussions for the other angles.



You can typically achieve two of the previous goals, but you will have to compromise on the third. For example, if your users' needs are met with a lower performance viewing experience, you can choose a protocol that requires lower bandwidth and lower computing power. However, if you must provide a high-performance viewing experience, you must have either higher bandwidth or higher computing power, and ideally both.

Each available display protocol handles the corners of the protocol triangle differently; each has its benefits and its drawbacks. When picking one or more display protocols, determine which protocol characteristics you need, and which trade-offs you can accept.

The following table is a subset of the display protocols that Leostream supports, and some of the defining characteristics of those protocols.

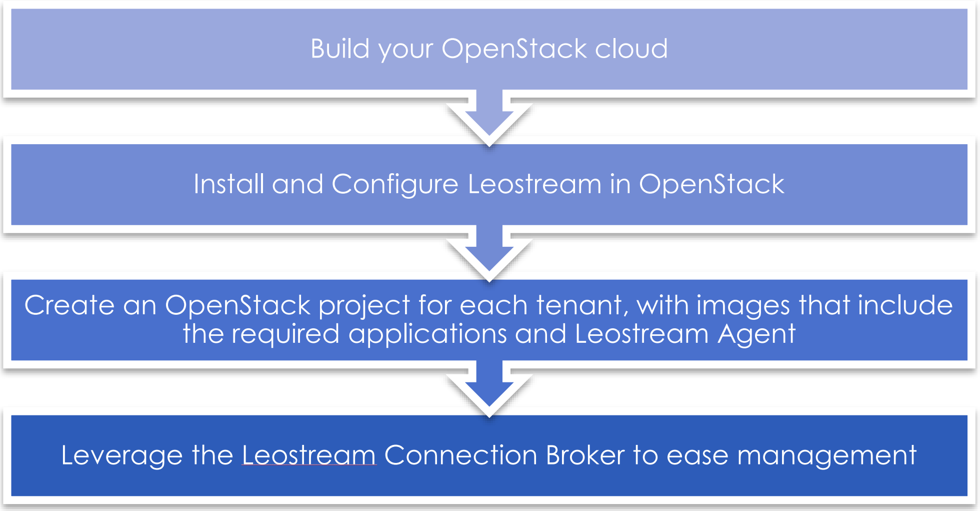
Display Protocol	Description
RDP	Using RDP does not require additional licensing fees in a Windows environment, so may help save money. The protocol continues to improve, but users running graphics-rich applications or videos may see inadequate performance. Many mobile device types have RDP clients that can be used in conjunction with Leostream, opening up access to a wider range of devices. RDP does not include a proxy solution, but Microsoft Direct Access may be an alternative for providing access into private OpenStack networks if floating IP addresses cannot be assigned to instances.

Display Protocol	Description
HP RGS	HP Remote Graphics Software (RGS) is a high-performance protocol, providing connections to graphics-rich applications. Please, contact HP for more information on pricing. RGS supports both Windows and Linux desktops. The Leostream Gateway supports RGS connections.
Mechdyne TGX	TGX delivers high resolution without sacrificing image quality or impacting performance. TGX supports both Windows and Linux desktops. End users can launch TGX connections from either the Leostream Connect client or using the Leostream Web client. The Leostream Gateway supports TGX connections.
Teradici PCoIP	The Teradici Cloud Access Software allows you to deliver virtual workspaces from your OpenStack cloud using the powerful PCoIP technology. The platform includes a built-in Security Gateway that can tunnel traffic from the outside world into a private OpenStack network. Please contact Teradici for more information on pricing.
NoMachine	NoMachine supports connections to Windows and Linux operating systems. It also allows you to deliver Linux sessions to end users, sharing a single Linux desktop with multiple users.
HTML5-based solutions	The Leostream Gateway supports HTML5-based RDP, VNC, and SSH connections, allowing you to provide access to a private OpenStack network without investing in other VPN solutions. HTML5-based connections can be used in any Web browser that supports HTML5. See the **Leostream Gateway Guide** for more information.

For more information on all the display protocols supported by Leostream, please consult the Leostream **Guide for Working with Display Protocols**, available on the Leostream website.

Implementing a Proof-of-Concept Environment

Implementing a proof-of-concept OpenStack VDI or DaaS environment consists of the following high-level steps.



Build your OpenStack cloud

Install and Configure Leostream in OpenStack

Create an OpenStack project for each tenant, with images that include the required applications and Leostream Agent

Leverage the Leostream Connection Broker to ease management

Building Your OpenStack Cloud

How you build your OpenStack cloud depends on the scale of your deployment, as well as other factors. There are a number of reference architectures available, as well as OpenStack experts who can help you design your OpenStack cloud.

This reference architecture focuses on aspects of your OpenStack cloud that you should pay particular attention to when integrating Leostream to manage OpenStack VDI or DaaS. Please, consult your OpenStack expert for details on building your OpenStack cloud.

Required User Permissions

Leostream manages OpenStack clouds using the OpenStack APIs. Before you can manage VDI or DaaS using the Leostream Connection Broker, ensure that any project you plan to use with Leostream has a user account with the required permissions in OpenStack to use the necessary APIs.

In order to use all of the functionality in the Connection Broker, your user requires access to the following:

In `/etc/nova/policy.json`

```
compute:get_all
compute:create
compute:start
compute:stop
compute:reboot
compute_extension:admin_actions:suspend
compute_extension:floating_ips
```

```
compute_extension:admin_actions:resume
network:allocate_floating_ip
network:associate_floating_ip
network:disassociate_floating_ip
```

In `/etc/glance/policy.json`

```
get_images
```

Leostream Security Groups

The security group assigned to your Leostream Connection Broker instance in OpenStack must open the appropriate ports for incoming traffic. You can define a new security group to open the ports required by the Connection Broker. However, the Leostream Connection Broker always assigns the default security group to new instances provisioned by Leostream. Therefore, ensure that you open the ports required by the Leostream Agent and display protocol on the **default** security group.

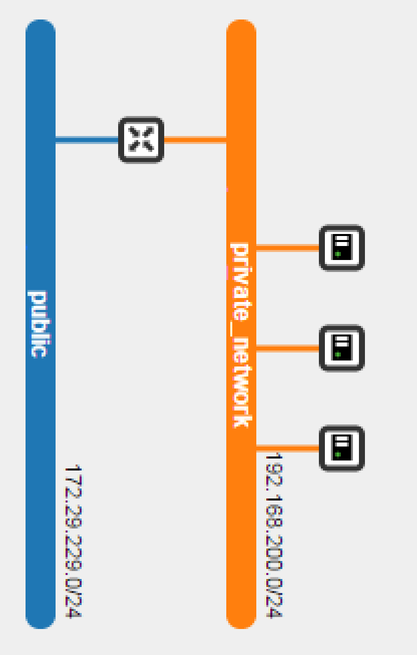
The following table describes the ports needed by the different components in the Leostream solution. All ports are TCP ports opened in the ingress direction.

Port	Required By	Purpose
22	Connection Broker	For SSH access to the Connection Broker. Alternatively, you can access the Connection Broker console via the Horizon Dashboard.
80 and 443	Connection Broker	For access to the Connection Broker Web interface, and communication with the Leostream Agents and Leostream Connect. If you close port 80 on your Connection Broker, you may omit that port from the security group.
8080*	Leostream Agent on the OpenStack Instances	Port for communications from the Connection Broker to the Leostream Agent. * The Leostream Agent port may be changed using the Leostream Agent Control Panel dialog. If you change the default Leostream Agent port, ensure that you open the associated port in the security group.
3389**	Display protocol on OpenStack Instances	For RDP access to the OpenStack VDI/DaaS instances ** If you use a display protocol other than RDP, ensure that you open any ports required by that display protocol.

Network considerations

When configuring your network for use in a VDI/DaaS environment managed by Leostream, you need to consider if your instances will be accessible to the external network, and take into account if you want Leostream to join new instances to your Active Directory domain.

This reference architecture uses a network structure that has a private and public network, as shown in the following figure.

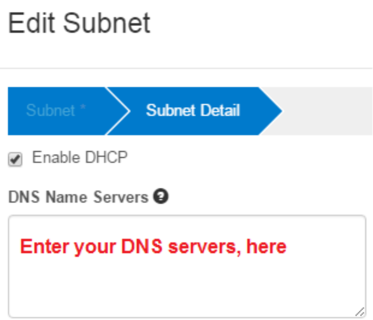


Floating IP addresses are assigned on the public network, and can be accessed outside of the OpenStack environment. The private network is internal to OpenStack.

Configuring DNS servers

If you plan to provision new instance using Leostream, and want Leostream to join those new instances to your domain, your new instances must be able to resolve the domain name. Ensure that the subnet where you place the instance uses a DNS server that can resolve your domain name. To set the DNS servers for your subnet, go to the **Network** page for your project and select your network. In the **Network Overview** page, edit the subnet.

You can set the **DNS Name Servers** in the **Subnet Details** page of the **Edit Subnet** form, for example:



Edit Subnet

☑ Enable DHCP

DNS Name Servers

Enter your DNS servers, here

Ensure that you can add an instance to your domain from within the desktop’s operating system before using Leostream to automate adding instances to your domain.

Isolating Instances on a Private Network

If you plan to isolate your OpenStack instances inside of a private network, you must provide a proxy solution, such as a VPN or security gateway, to tunnel the end user into the private network. Alternatively, if your users connect to their desktops using HP RGS or RDP, you can use the Leostream Gateway to proxy the traffic. The Leostream Gateway also provides HTML5 RDP connections, enabling in-browser connections. See the **Leostream Gateway Guide** for more details.

Some display protocol solutions, such as the Teradici Cloud Access Platform, provide a built-in security gateway that can tunnel the end-user's desktop connect into the private OpenStack network. Other display protocols, such as Microsoft RDP, require you to investigate alternative solutions, such as using Microsoft Direct Access.

If you provide the OpenStack instance with a floating IP address, Leostream can use that address to establish desktop connections for some display protocols. Please note that some display protocols, such as PCoIP, require a proxy solution into the internal network if the desktops DNS name resolves to the private IP address.

Metadata

The Leostream Agent installed on the OpenStack instances uses the metadata to retrieve information about the instance, such as its public IP address. You must enable metadata in your OpenStack cloud and ensure that new instances automatically include a route to the 169.254.169.254 address.

On a Microsoft Windows instance, you can use the `route PRINT` command to ensure that a route to the metadata address exists.

If the route does not exist, you can use the `route` command to add the appropriate route, for example:

```
route -p add 169.254.169.254 mask 255.255.255.255 192.168.200.1 metric 6
```

Installing the Leostream Connection Broker

The Connection Broker and Leostream Gateway are provided as packages that can be installed on a CentOS, Red Hat Enterprise Linux, Ubuntu, or SUSE Linux Enterprise operating system. The Connection Broker and Leostream Gateway must be installed on separate machines.

See the **Leostream Installation Guide** for complete instructions.

Configuring OpenStack Images for VDI

This reference architecture assumes that you have a base operating system image that can be used to launch instances in your OpenStack cloud. When working with KVM, Microsoft Windows instances must be prepped with the appropriate drivers. You may start with your own version of Windows, or may find it easier to use a preconfigured image, such as the Windows Server 2012 image provided by Cloudbase. The OpenStack Community App Catalog contains a number of preconfigured images for Linux operating systems.

Supported Operating Systems

The Leostream Connection Broker can manage connections to OpenStack instances running a Windows or Linux operating system, including:

- Windows Server 2008 and Windows Server 2008 R2
- Windows Server 2012 and Windows Server 2012 R2
- Windows Server 2016
- Windows 7, including SP1
- Windows 8 and 8.1
- Windows 10

- CentOS
- Debian
- Fedora
- SUSE Linux Enterprise
- Red Hat Enterprise Linux
- Ubuntu

When creating instances within the Horizon Dashboard, ensure that you install the appropriate Leostream Agent on the instance and register that agent with your Leostream Connection Broker, as described in the following section. Any images that you want to use within Leostream to provision new instances must also include the Leostream Agent.

Installing the Leostream Agent

When installed on a desktop, the Leostream Agent provides the Connection Broker with additional information about the user’s session, including:

- When the user logs into the remote desktop
- When the user disconnects from the remote session
- When the user logs off of the remote desktop
- When the user locks or unlocks their remote desktop
- When the user’s session is idle

On a Windows operating system, the Leostream Agent also performs tasks required to join new instances to an Active Directory domain.

Leostream provides a Leostream Agent for Windows operating systems, and a Java version of the Leostream Agent for Linux operating systems. Ensure that you download the appropriate Leostream Agent from the Leostream Downloads page. Consult the Leostream Installation Guide for instructions on how to install the Leostream Agent on your OpenStack instances.

The Connection Broker address can be specified when you install the Leostream Agent. If you need to specify or change the Connection Broker address after the Leostream Agent is installed, you can use the Leostream Control Panel dialog in Windows or set the address in the leostreamagent.conf file on Linux. See the Leostream Agent Administrator's Guide for more information.

Microsoft Licensing Considerations

If you are building a VDI solution on OpenStack, you can host any available version of a Microsoft Windows operating system. When delivering desktops-as-a-service from an OpenStack cloud, you typically must use a Windows Server operating system. Please, consult your Microsoft licensing expert for the more up-to-date and relevant information for licensing Windows in your OpenStack environment.

Joining Desktops to your Domain

When building a base Windows instance for Leostream to use to provision new desktops, ensure that you do not join that instance to your Active Directory domain. Leostream only adds desktops to a domain if that desktop is currently a member of a Workgroup. Leostream will not move desktops to different domains, or change the hostname of a desktop that is already joined to a domain. Therefore, before you create an image of the instance, ensure that the original instance is part of a local Workgroup.

As part of joining a new instance to your domain, the Leostream Agent can change the desktops hostname, so you can ensure that you do not have duplicate hostnames in your environment. If you need to perform additional sysprep steps on a Windows instance, you can use `cloudbase-init` to initialize the instances in your OpenStack environment. Please, visit http://www.cloudbase.it/cloud-init-windows/ for more information on using cloud-init.

You can also use `cloud-init` to initialize Linux instances in your OpenStack environment.

Connecting Leostream to your OpenStack Cloud

This section includes information on connecting your Leostream Connection Broker to your OpenStack cloud, and using Leostream to manage the lifecycle of OpenStack instances. For complete information on configuring Leostream pools, plans, policies, and assignments, see the Leostream Guide for Installing and Configuring Leostream in an OpenStack environment.

To connect Leostream to your OpenStack cloud, create an OpenStack *center* in Leostream.

*Leostream defines **centers** as the external systems that host the desktops and other resources that the Connection Broker manages and assigns to end users.*

Leostream uses the OpenStack APIs to inventory the instances and images in your OpenStack cloud. Ensure that you have a user account that has the appropriate permissions for the OpenStack project you plan to use in your Connection Broker (see Required User Permissions).

To manage instances in multiple projects, create a center for each project. To create an OpenStack center:

1. Go to the **> Setup > Centers** page.
2. Click the **Add Center** link.
3. In the **Add Center** form, select **OpenStack** from the **Type** drop-down menu.

   If you do not see OpenStack in your list of provided options, please contact **sales@leostream.com** to update your Leostream license key.

4. Enter a name for the center in the **Name** edit field.
5. In the **Auth URL** edit field, enter the public URL to the OpenStack Keystone identity service endpoint, for example:

   ```
   http://external_openstack_ip:5000/v3.0
   ```

   Where *external_openstack_ip* is the externally accessible IP address to your identity service.

   Leostream supports only version 3 of the Keystone API.

6. Enter the OpenStack domain that contains your project and user in the **Domain** edit field.
7. Specify the project you want to manage in the **Project** edit field.
8. In the **Username** edit field, enter the name of a user with the necessary permissions for this project.
9. Enter this user's password into the **Password** edit field.
10. Click **Save** to create the center.

The instances in the center’s OpenStack project appear on the Connection Broker **> Resources** > **Desktops** page. The Connection Broker inventories all images and displays them on the **> Resources > Images** page. See the “Working with Desktops and Applications” section of the Connection Broker Administrator’s Guide for information on viewing, editing, and controlling desktops from within the Connection Broker.

Building a Pool

To support multi-tenancy and ease administration, group OpenStack instances into pools based on OpenStack projects.

The Leostream Connection Broker defines a ***pool*** *as any group of desktops or applications*

An instance’s pool membership determines what users have access to the desktop and how Leostream manages the desktop connection. Pool configurations also determine when Leostream launches new instances in the project, to expand your compute capacity.

To create a pool that contains all of the instances in a particular OpenStack project:

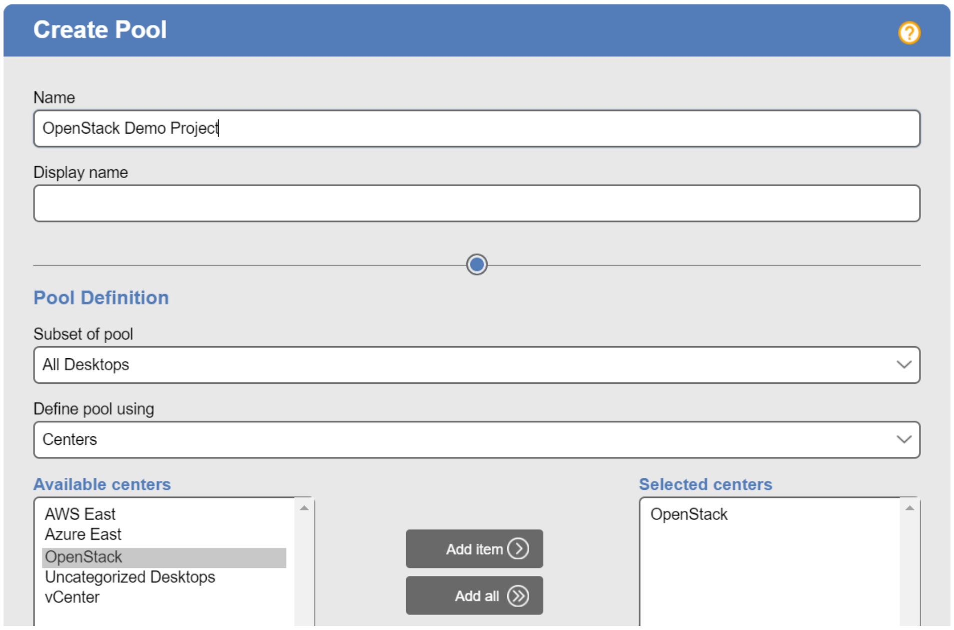
1. Go to the **> Configuration > Pools** page.

2. Click **Create Pool**.

3. Enter a unique name for this pool in the **Name** edit field.

4. Select **Centers** from the **Define pool using** drop-down menu.

5. In the **Center Selection** section, select the appropriate OpenStack center from the **Available centers** list, for example:



Create Pool

Name: OpenStack Demo Project

Display name:

Pool Definition

Subset of pool: All Desktops

Define pool using: Centers

Available centers: AWS East, Azure East, OpenStack, Uncategorized Desktops, vCenter

Selected centers: OpenStack

6. Click the **Add item** button.

7. Click **Save**.

For a complete description of pools, see the “Creating Desktop and Application Pools” chapter in the Connection Broker Administrator’s Guide.

The pool created in this example contains all of the instances already in the project associated with the selected OpenStack center. To instruct the Connection Broker to launch new instances, configure the **Provisioning Limits** and **Provisioning Parameters** sections in the **Edit Pool** page, as described in the next section.

Provisioning New OpenStack Instances

Your Leostream license determines if provisioning is enabled in your Connection Broker. If you do not see the options described in this section, contact **sales@leostream.com** to update your license key.

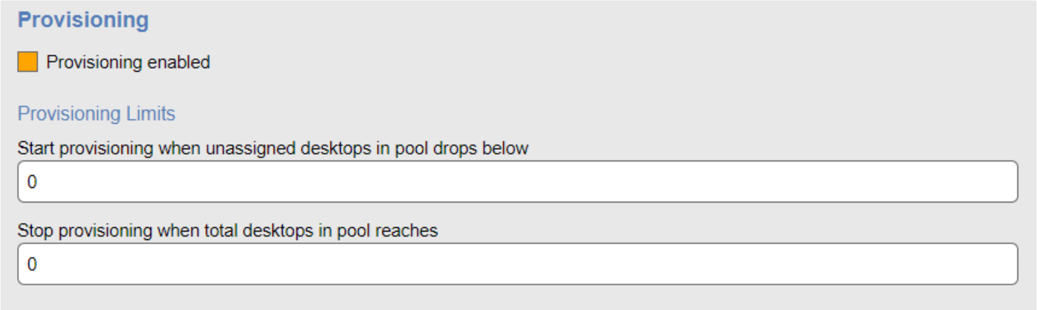
Provisioning allows you to generate new OpenStack instances when the number of desktops in a pool reaches a specified lower threshold. Before provisioning instances in an OpenStack environment, you must configure the following:

1. Create master images. Your available images are displayed on the **> Resources > Images** page. Ensure that your master images contain an installed Leostream Agent and that agent is configured to communicate with your Connection Broker.

2. Configure a network in the OpenStack project. Ensure that the network ID for this network is included in the **Network UUID** field of your OpenStack center.

If you do not properly configure a network, the Connection Broker cannot provision new instances in OpenStack.

The **Provisioning** section of the **Edit Pool** page allows you to configure when and how the Connection Broker creates new instances in your OpenStack environment. By default, the **Provisioning enabled** checkbox is selected, as shown in the following figure, and provisioning is on for all your pools.



Provisioning

☑ Provisioning enabled

Provisioning Limits

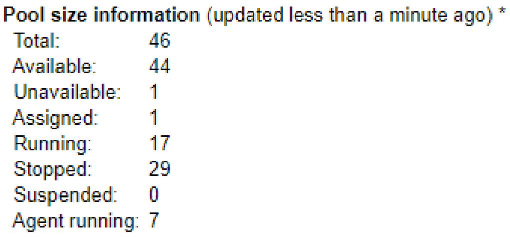
Start provisioning when unassigned desktops in pool drops below

0

Stop provisioning when total desktops in pool reaches

0

The Connection Broker determines when to create new instances by comparing the thresholds specified in the **Provisioning Limits** section to the current contents of the pool. If you edit an existing pool, the Connection Broker displays the current contents of the pool size to the right of the **Edit Pool** form, for example:



Pool size information (updated less than a minute ago) *

Total:	46
Available:	44
Unavailable:	1
Assigned:	1
Running:	17
Stopped:	29
Suspended:	0
Agent running:	7

The number entered into the **Start provisioning when unassigned desktops in pool drops below** field specifies a lower bound on the number of unassigned desktops in the pool, where the number of unassigned desktops is the total number of desktops minus the number of assigned desktops.

For example, the previous figure shows one assigned desktop and 46 total desktops. Therefore, there are 45 unassigned desktops. An unassigned desktop can have a desktop status of either available or unavailable.

The Connection Broker checks the provisioning limits, and creates new instances, at the following times

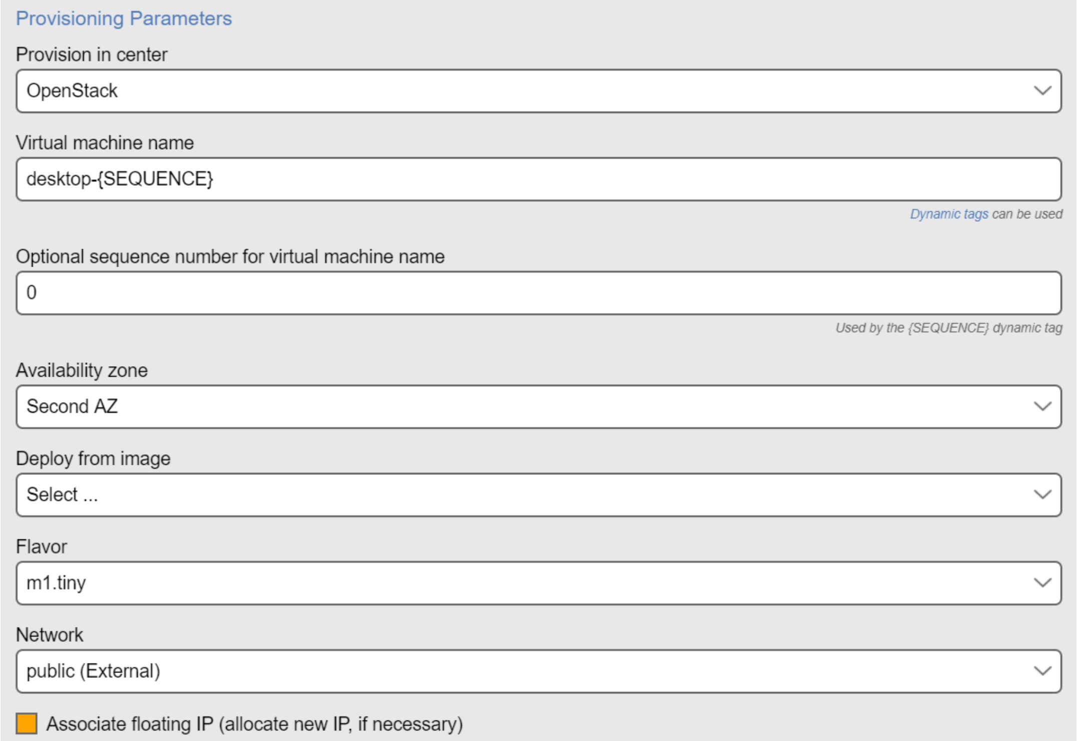
- When the pool is saved
- When a user is assigned to a desktop in this pool

- When any `pool_stats` or `pool_history_stats` job runs

The Connection Broker continues to provision new desktops whenever the lower threshold is crossed, until the upper threshold specified in the **Stop provisioning when total desktops in pool reaches** field is reached, indicated by the **Total** value in the pool size information.

After defining provisioning limits, use the **Provisioning Parameters** section to configure provisioning in OpenStack:

1. From the **Provision in center** drop-down menu, select the OpenStack center, and therefore project, to provision new machines into. The remainder of the form updates based on the contents of your selection. The following figure shows a subset of the **Provisioning Parameters**.



Provisioning Parameters

Provision in center: OpenStack

Virtual machine name: desktop-{SEQUENCE}

Dynamic tags can be used

Optional sequence number for virtual machine name: 0

Used by the {SEQUENCE} dynamic tag

Availability zone: Second AZ

Deploy from image: Select ...

Flavor: m1.tiny

Network: public (External)

□ Associate floating IP (allocate new IP, if necessary)

2. Enter a name for the virtual machine in the **Virtual machine name** edit field. You can use dynamic tags to create a name from a mixture of static and dynamic variables. See "Using Dynamic Tags in Provisioning Parameters" in the Connection Broker Administrator's Guide for an example.

3. If the virtual machine name contains a `{SEQUENCE}` dynamic tag, enter the starting number for the sequence in the **Optional sequence number for virtual machine name** edit field. The Connection Broker increments the number for each new instance.

4. Select the availability zone to provision the new instance into from the **Availability zone** drop-down menu.

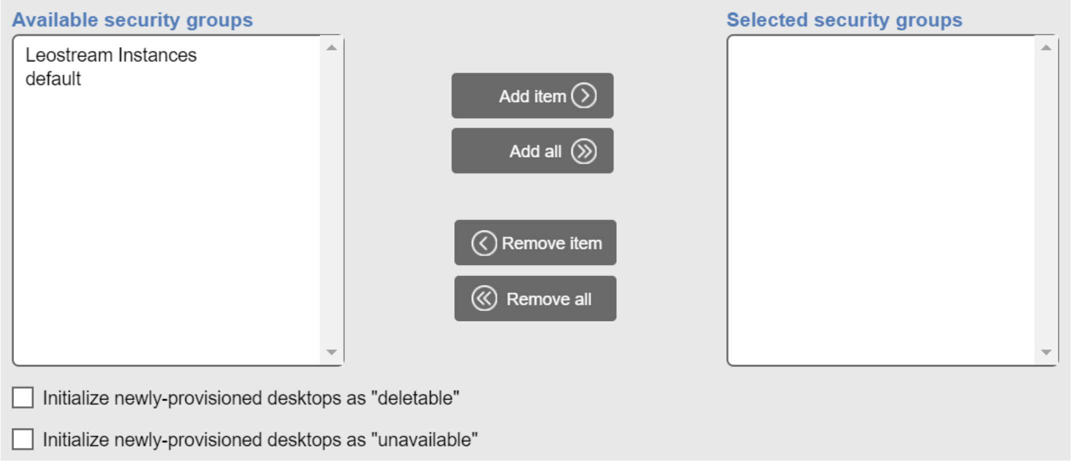
5. Select the image to use from the **Deploy from image** drop-down menu. This menu contains all the public and project images available in the OpenStack center you selected.

6. Select the instance size from the **Flavor** drop-down menu.

7. Select a network location for the instances from the **Network** drop-down menu.

8. Select the **Associate floating IP (allocate new IP, if necessary)** option if Leostream should assign a floating IP address to the new instance. If a floating IP address is not available, Leostream attempts to allocate a new address.

9. In the **Available security groups** field, select the security groups to assign to the new instance. Click the **Add item** button to place them into the **Selected security groups** field.



□ Initialize newly-provisioned desktops as "deletable"

□ Initialize newly-provisioned desktops as "unavailable"

10. Select the **Initialize newly-provisioned desktops as deletable** option to indicate that the Connection Broker is allowed to terminate this instance from OpenStack. When this option selected, the **Edit Desktop** page for the newly provisioned desktop has the **Allow this desktop to be deleted from disk** option selected, by default. Use release plans to schedule instance deletion.

11. Select the **Initialize newly provisioned desktop as unavailable** option to set the desktop status to `Unavailable`. The Connection Broker will not offer a desktop to users if the desktop's status is set to `Unavailable`, allowing you to perform post-provisioning actions on the desktop.

12. Click **Save**.

When the number of unassigned desktops in the pool falls below the lower threshold, the Connection Broker creates a new instance from the selected image. Desktops that are marked as deletable can be assigned a release plan that terminates the desktop after the user logs out. This allows you to model the single-use desktops described in **Storage**. For information on configuring Leostream release plans for single-use or permanent desktops, see the Leostream Quick Start Guide for OpenStack VDI and Desktops-as-a-Service**.**

Production Deployments

The following two sections include information on moving from a proof-of-concept to production environment. For more information on deploying Leostream at scale, please see the Connection Broker Administrator's Guide.

Scaling and High Availability

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 data center goes down, users are able to log in to resources in a disaster recovery data center.

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.

A Connection Broker *cluster* is a group of Connection Brokers that share the same PostgreSQL or Microsoft SQL Server® 2012, 2014, or 2016 database. A common cluster uses two to three 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 data centers 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.

Database Requirements

In order to share information between Connection Brokers in a cluster, you must use an external database. The Connection Broker supports PostgreSQL and Microsoft SQL Server 2012, 2014, and 2016 database servers.

Database Space Requirements

The Connection Broker uses the database to store all logs and information about each center, desktop, user, etc. Each desktop and unique user requires approximately 1KB of storage space. Every user login and logout creates 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" in the Connection Broker Administrator's Guide 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 single 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.

Distributing User Logins

When moving to a clustered Connection Broker environment, use a load balancer to spread user connections around the clustered Connection Brokers. All 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.

Your DNS server provides an inexpensive method for distributing user connections between Connection Brokers in a cluster and can allow you to meet your system capacity requirements. Using DNS, you can *regionalize* your Connection Broker, i.e., when a user logs into the Broker, they have access to the local DNS

⚠ *A simple DNS system cannot detect failure of a single Connection Broker host, and continues to hand that Connection Broker address to users. A user assigned to a failed Connection Broker address must wait until the connection times out before another Connection Broker address is tried. Therefore, using DNS for load balancing is suitable only for systems that can stand a moderate amount of delay during failover.*

name server and, hence, the local Connection Broker. You can override this regional behavior, i.e., send your users to their home Connection Broker, using the Connection Broker's user redirection feature.

Conclusions

OpenStack software is quickly becoming a viable alternative for managing large-scale data centers. Using Leostream, you can host VDI and desktops-as-a-service in your OpenStack cloud alongside other servers, applications, and use cases. By utilizing open source software to manage your cloud, you avoid the costly licensing fees associated with full virtualization stacks. By using the Leostream Connection Broker, you can optimize resource usage, and lower costs even further by providing users with Linux desktops, as well as Microsoft Windows desktops. In the end, with Leostream and OpenStack, you turn your cloud into a VDI powerhouse that can satisfy any end-user need.

Additional Resources

- **Leostream Connection Broker Free Trial**
- **Leostream Connection Broker OpenStack VDI solutions page**
- **eBook: Building OpenStack VDI and DaaS**
- **Webinar: How to make OpenStack VDI and DaaS a Reality**
- **Infographic: How to Manage OpenStack VDI with Leostream**
- **Leostream Quick Start Guide for OpenStack VDI and DaaS**
- **Leostream Downloads page for Leostream Agents and Leostream Connect clients**