Aspera® Proxy 1.0.1

Red Hat, Debian

ADMIN GUIDE

7 Nov 2013

Technical Support

Contact Info

Email	support@asperasoft.com
Phone	510-849-2386
Request Form	**http://support.asperasoft.com/home**

Support Service Hours

Standard Support	8:00am – 6:00pm Pacific Time (GMT-8)
Premium Support	8:00am – 12:00am Pacific Time (GMT-8)

Support Closed

Weekends	Saturday, Sunday
Aspera Holidays	Please see **http://support.asperasoft.com/home**

Feedback

The Aspera Technical Publications department wants to hear from you on how Aspera's user manuals can be improved. To submit feedback about this manual, or any other Aspera product document, please visit the **Aspera Product Documentation Feedback Forum**. Through this forum, you can let us know if you find content that is unclear or appears incorrect. We also invite you to submit ideas for new topics, as well as ways that we can improve the documentation to make it easier for you to read and implement. When visiting the Aspera Product Documentation Feedback Forum, please remember the following:

- You must be registered to use the Aspera Support Website at:

  https://support.asperasoft.com

- Be sure to read the forum guidelines before submitting a request.

Corporate Headquarters
Aspera, Inc.
5900 Hollis Street
Suite E
Emeryville, CA 94608
U.S.A.

T: 510-849-2386
F: 510-868-8392

Aspera Proxy 1.0.1 Admin Guide
Red Hat, Debian

7 Nov 13

www.asperasoft.com

C. Troubleshooting

Aspera Proxy 1.0.1

Admin Guide

1. Introduction

Welcome to Aspera Proxy, Aspera's open and authenticated proxy solution built on top of the Linux kernel. Proxy protects your organization's network while enabling secure, high-speed *fasp* transfers to and from highly restrictive network environments. It allows transparent pass-through of *fasp* transfer sessions across secure DMZs without impeding transfer speeds or compromising the security of your internal network.

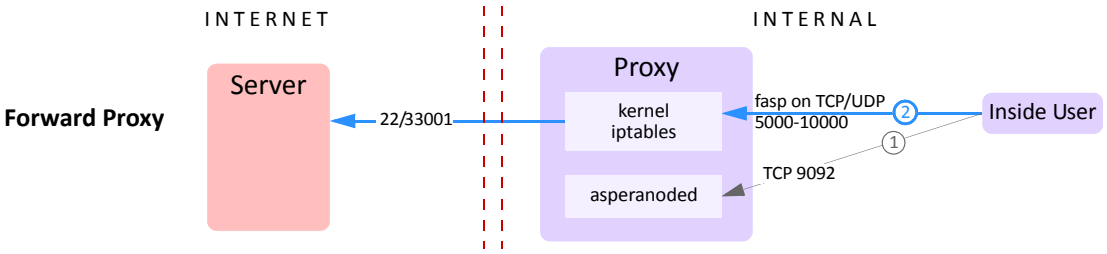
Aspera Proxy also supports load balancing, high availability, and flexible security policies. It consolidates *fasp* transfers in and out of a corporate network and enables precise control over which users can initiate transfers with remote Aspera transfer servers. With Proxy support built into all Aspera desktop and browser-based transfer clients, its configuration and use is straightforward for all your users.

Aspera Proxy supports both forward (outbound) and reverse (inbound) proxy modes, allowing *fasp* transfers to be initiated by users who are either inside or outside the corporate network.

Forward Proxy

Forward proxy provides a secure way for users behind company network firewalls to initiate requests for *fasp* transfers of files that are on servers outside the firewall. It addresses the following customer use cases:

- **Limited-use Internet access**: Your enterprise has security requirements that prevent you from deploying the Aspera Enterprise Server (or Connect Server) inside your DMZ. Organizations often limit general Internet access for their employees, which can affect the *fasp* protocol even if used for legitimate business needs. Aspera Proxy provides secure access to the Aspera transfer servers residing outside of your corporate network without exposing users' IP addresses. It also enforces strict user authentication for Aspera clients that initiate connections to the outside servers.
- **Consolidation and Control of fasp transfers**: If you are an IT systems manager and want to establish better control and security around *fasp* transfers that your internal users initiate, Aspera Proxy can fulfill your requirements without impeding the users' experience. It provides a single point through which all *fasp* transfers flow in and out of your corporate network, hiding internal clients' IP addresses and allowing you to control which users can initiate *fasp* transfers, without slowing down the speed of the transfers.



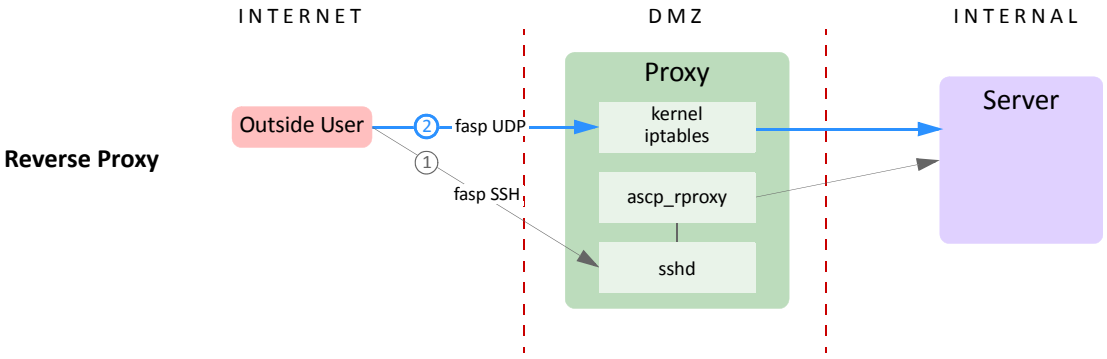
Reverse Proxy

Reverse proxy provides a secure way for users outside company network firewalls to initiate requests for *fasp* transfers from servers inside the firewall. It addresses the following customer use case:

- **Trusted partners need access to files on your servers**: Customers want to allow users outside their company firewall to initiate *fasp* transfers to and from servers inside the company network.

Reverse proxy is usually deployed inside a DMZ, on top of a Linux-based server. Multiple proxy instances can also be launched on a server cluster, behind an enterprise-grade load balancer, forming a high-availability solution. Reverse proxy currently employs the same security model as Aspera's Connect Server or Enterprise Server, based on the SSHD service. As a result, no changes are needed on the client side. Once authenticated, the proxy server invokes one program: **ascp_rproxy**, which is in charge of bidirectional forwarding of SSH control traffic and *fasp* (UDP) traffic between the client and the internal server.

The **ascp_rproxy** program proxy server maintains an SSH connection with the **ascp** client when it's invoked by the SSHD service. A second SSH connection is set up between the proxy server and the internal Enterprise Server instance by virtue of a pre-installed SSH key. It then bridges the two SSH connections, by forwarding incoming data from one connection to the other, in both directions. In order to forward *fasp* (UDP) traffic, the **ascp_rproxy** program proxy server sets up a dynamic network address translation (DNAT) rule using the Linux **iptables** kernel module. Since UDP traffic forwarding is done using the Linux **iptables** kernel module, high-speed packet forwarding can be achieved without any reduction in speed.



Hardware and Software Requirements

The use of Aspera Proxy requires the following:

- A Linux system (Red Hat or Debian) with kernel 2.4+.
- iptables v1.3.0+ installed and not blocking TCP/UDP 33001.

NOTE: Do not install Aspera Proxy on a machine where Aspera Enterprise Server or Connect Server is installed. If these products are already installed, be sure to remove them before installing Aspera Proxy.

2. Installation

To install Aspera Proxy, log into your computer with root permissions, and follow the steps below.

Step 1 **Download the Aspera product installer.**

Download the Aspera Proxy installer from the link below. To access, use the credentials Aspera has provided to your organization:

http://downloads.asperasoft.com/en/downloads/42

If you need help determining your firm's access credentials, contact Aspera Technical Support.

Step 2 **Run the installer.**

Once downloaded, run the installer using the following commands and with the proper administrative permissions:

Red Hat Linux:
```
# rpm -Uvh aspera-proxy-version.rpm
```

Debian-based Linux:
```
# dpkg -i aspera-proxy-version.deb
```

This starts the Aspera proxy daemon, and makes adjustments to the **iptables** system settings.

Step 3 **Install the license.**

In a terminal window, create the following file and paste your license key string into it:

/opt/aspera/proxy/etc/aspera-license

If you're updating an existing license, simply open the file and replace the existing license string with a new one.

When finished, save and close the file. Run the following command to verify the license info:

```
# ascp -A
```

This lets you know whether Aspera Proxy is correctly installed.

Step 4 **Review or update OpenSSH authentication methods.**

Open your SSH server configuration file with a text editor:

/etc/ssh/sshd_config

To allow public key authentication, set **PubkeyAuthentication** to **yes**. If you also plan to allow password authentication, which is less secure than keys, set **PasswordAuthentication** to **yes**:

```
...
PubkeyAuthentication yes
PasswordAuthentication yes
...
```

NOTE: For information about security options with Aspera products, see *Appendix B: Securing your SSH Server*

Then, execute the following command to restart SSH:

Red Hat Linux:
```
# service sshd restart
```

Debian-based Linux:
```
# /etc/init.d/ssh restart
```

Step 5 **Adjust settings in your Linux environment.**

1. IP forwarding must be enabled and is enabled automatically when Aspera Proxy is installed. To confirm, run the following command:

```
$ cat /proc/sys/net/ipv4/ip_forward
```

If the command returns 1, IP forwarding is enabled. If it returns 0, it is not. IP forwarding can be enabled manually by setting the **net.ipv4.ip_forward** line in **/etc/sysctl.conf** as follows:

```
# Controls IP packet forwarding
net.ipv4.ip_forward=1
```

To activate changes to **/etc/sysctl.conf**, run the following:

```
$ /sbin/sysctl -p /etc/sysctl.conf
```

2. Verify that the following entry is present in **/etc/hosts**:

```
127.0.0.1 localhost
```

3. Ensure that SELinux is disabled. **SELINUX** can be set to "permissive" or "disabled," but not "enforced." Configuration is done in **/etc/sysconfig/selinux** or **/etc/selinux**, if present.

3. Forward Proxy

Configuring the Proxy Server

The configuration steps below are to be entered in the **aspera.conf** file provided with your Aspera Proxy distribution and found in the following location:

/opt/aspera/proxy/etc/aspera.conf

Step 1 **Enable HTTP and/or HTTPS in aspera.conf.**

In **aspera.conf**, set **<enable_http>** and/or **<enable_https>** to true. These settings enable HTTP and HTTPS for the node API services.

Add (or update) the **<server>** section as follows and set the **<enable_http>** and/or **<enable_https>** options to true.

```
<server>
  ...
    <enable_http>true</enable_http>        <!-- true | false -->
    <enable_https>true</enable_https>      <!-- true | false -->
  ...
</server>
```

Keep **aspera.conf** open for the next step.

Step 2 **Add the <proxy> section to aspera.conf.**

In **aspera.conf**, copy and paste the following **<proxy>** section into the file's **<server>** section:

```
<server>
  ...
    <proxy>
      <enabled>true</enabled>         <!-- Proxy server is enabled -->
    </proxy>
  ...
</server>
```

Within the **<proxy>** section, **<enabled>** is set to true. In general, this the only option you need to set on the proxy server in order to begin using forward proxy; however, you may need to change other **<proxy>** settings based on your unique network configuration.

Step 3 **Update additional forward proxy settings, as needed.**

To view all forward-proxy configuration options, run the **asuserdata** command as follows:

```
$ /opt/aspera/proxy/bin/asuserdata -s
```

NOTE: The **asuserdata -s** command displays the *default* values for the server setup, *not* the currently-set values.

After running this command, scroll down to the **<!-- Server Options Spec -->** section. All configuration options for the forward proxy server are displayed here in the **<proxy>** subsection:

```
<!-- Server Options Spec -->
...
  <proxy>
    <enabled>false</enabled>                          <!-- proxy_enabled: boolean true|false -->
    <authentication>false</authentication>            <!-- proxy_authentication: true|false -->
    <bind_ip_address>0.0.0.0</bind_ip_address>        <!-- proxy IP address: IP address -->
    <bind_ip_netmask></bind_ip_netmask>               <!-- proxy IP netmask: blank by default -->
    <port_range_low>5000</port_range_low>             <!-- proxy port range lower bound: integer -->
    <port_range_high>10000</port_range_high>          <!-- proxy port range upper bound: integer -->
    <cleanup_interval>0</cleanup_interval>            <!-- proxy cleanup interval: integer -->
    <keepalive_interval>0</keepalive_interval>        <!-- proxy keep-alive interval: integer -->
    <session_timeout>0</session_timeout>              <!-- proxy session timeout: integer -->
  </proxy>
  ...
```

Setting	Description	Default Value
<enabled>	Disable or enable the proxy server. Must be set to true to turn on the service.	false
<authentication>	Disable or enable the authentication requirement for the proxy server.	false
<bind_ip_address>	The IP address that the proxy server binds to (also the IP address that the client connects to). The default value, 0.0.0.0, allows the proxy server to bind to all available interfaces.	0.0.0.0
<bind_ip_netmask>	The netmask that the proxy server binds to (also the netmask that the client connects to).	blank (null)

Setting	Description	Default Value
<port_range_low>	The lower bound of the port range. Ensure that the firewall allows the port you specify.	5000
<port_range_high>	The upper bound of the port range. Ensure that the firewall allows the port you specify.	10000
<cleanup_interval>	The interval, in seconds, at which the proxy server scans and cleans up expired sessions.	0
<keepalive_interval>	The interval, in seconds, after which a session times out if no keep-alive updates have been received.	0
<session_timeout>	The interval, in seconds, at which an **ascp** client sends keep-alive requests. This option is propagated to the client.	0

Whenever you make changes to **aspera.conf**, you can validate the syntax and tags by running **asuserdata** with the **-v** option:

```
$ /opt/aspera/proxy/bin/asuserdata -v
```

Step 4 **Restart the proxy node service.**

After modifying **aspera.conf**, save it and restart the proxy node service as follows:

```
$ sudo /etc/init.d/asperaproxy restart
```

If you receive the following error when attempting to start the node service, check to see if **iptables** is installed on your machine:

```
ERR Failed to initialize proxy service
```

If **iptables** is not installed, run the following command (based on your Linux distribution):

Red Hat Linux:

```
$ sudo yum install iptables
```

Debian-based Linux:

```
$ sudo apt-get install iptables
```

Step 5 **Check log entries for startup.**

After starting up the **asperanoded** service, check the system log-file entries:

Red Hat Linux:

```
/var/log/messages
```

Debian-based Linux:

```
/var/log/syslog
```

The only proxy entries that should be displayed are the following:

```
LOG proxy service ready (port range 5000-10000)
LOG Started on port(s) 9091,9092s ...
```

The port range (lower and upper bounds) can be modified by changing the **<port_range_low>** and **<port_range_high>** options in the **<proxy>** section of **aspera.conf**; whereas, the default node

service ports (9091 and 9092) can be modified by changing the **<http_port>** and **<https_port>** options in the **<server>** section.

Step 6 **Create a node API user.**

On the proxy machine, create a node API user by running the **asnodeadmin** command:

```
$ sudo /opt/aspera/proxy/bin/asnodeadmin -au node_api_user -p password -x transfer_user
```

The *transfer_user* must be an existing user on the proxy server.

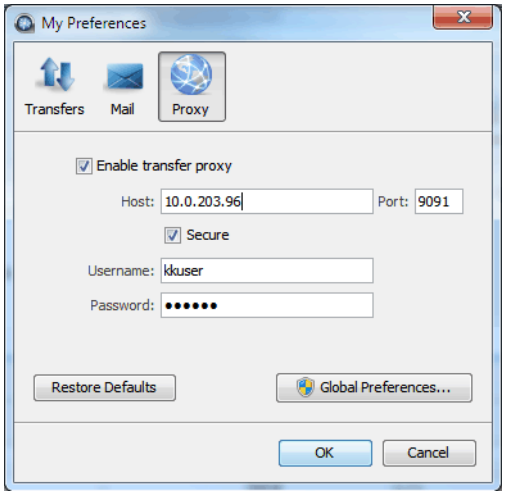
Firewall Considerations

Your Aspera transfer products require access through the ports you have designated for SSH and UDP, typically port 33001. If you cannot establish connections, review your local corporate firewall settings and ensure that restrictions on these ports are removed accordingly. For details, see *Appendix A: Configuring Firewalls*.

Configuring the Client

You configure your client transfer application by specifying your proxy host, port number, username, and password. In the Enterprise Server GUI, go to **Preferences > Proxy**. Note that the transfer proxy feature is disabled by default. On this screen, you can do the following:

- Configure connections on a case-by-case basis using this screen.
- Configure proxy settings for all transfers by clicking **Global Preferences**. This requires root privileges.



Case-by-Case Settings

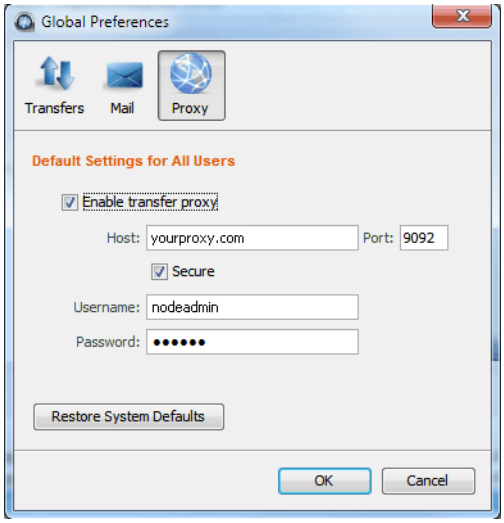
1. Check the **Enable transfer proxy** checkbox if you want to turn on transfer proxy and override global settings for connecting to your proxy server.

2. Enter the proxy server's hostname or IP address, and enter the port number.
3. Enable the **Secure** checkbox if your proxy server allows secure connections.
4. Enter the proxy server's node API username and password. This is the node API user you created above in Step 6 when you were configuring the proxy server.

Global Settings

Setting global preferences for proxy transfers requires root/admin privileges. To configure your global proxy settings, click the **Global Preferences** button. In the **Global Preferences** window, fill out the choices as follows:

1. Check the **Enable transfer proxy** checkbox. Note that the transfer proxy facility is disabled by default.
2. Enter the proxy server's hostname or IP address, and enter the port number.
3. Check the **Secure** checkbox if your server requires secure connections (recommended).
4. Enter your proxy server's node API username and password. This is the node API user you created above in Step 6 when you were configuring the proxy server.



4. Reverse Proxy

Configuring External Clients

On external clients from which reverse-proxy Aspera transfers will be initiated, ensure that each user has generated an SSH key pair using **ssh-keygen**. The Linux default files and location for private/public key pairs is **id_rsa** and **id_rsa.pub** in **/home/*user*/.ssh/**. However, the defaults are not a requirement for Aspera Proxy.

You will install the generated public keys (**id_rsa.pub**) when you set up accounts on the proxy server.

Configuring the Proxy Server

Step 1 **Create users and generate SSH keys.**

There are two approaches to setting up proxy accounts:

- **Squashed user account**: Multiple users make transfers to a single "squashed" proxy account. This approach is less effort to set up, because individual accounts are not required on the internal destination server. However, its disadvantage is that when transferred files arrive at their destination, they are all owned by the squash-user. The squashed approach is generally considered the best choice for Faspex.
- **Individual user accounts**: Each user makes transfers through their own proxy account. The advantage is that each transferred file is still owned by the user who initiated the transfer when it arrives at its destination. The individual-account approach is generally considered the better choice for transfers initiated using Connect Server and Enterprise Server.

  Individual accounts are required on both the proxy server and the internal destination server.

Note that a proxy server can accommodate a mix of squashed and individual-account approaches. A proxy server is not restricted to just one approach. The following steps cover both approaches.

1. Log into the proxy server as root, and create an account for each user. Account names should correspond to the accounts that users have on external clients.

```
# adduser user
```

2. Generate an SSH key pair for each user:

```
# su - user -c ssh-keygen
```

   By default, the **ssh-keygen** command generates and copies the private key (usually **id_rsa**) and public key (usually **id_rsa.pub**) to the **.ssh** directory in the user's home directory, typically **/home/*user*/.ssh**.

   In later steps, you will specify the location of the private keyfile when you set up **aspera.conf**.

Step 2 **Set up public keys generated on the external client.**

1. For each user, create the file **authorized_keys** in **/home/*user*/.ssh**.
2. To each **authorized_keys** file, append the public key that was generated for that user on the external client node.

Step 3 **Grant sudo access to proxy users.**

Log in to the proxy server as superuser and modify the **/etc/sudoers** file as follows:

1. Add the following line for each account on an **ascp** client that will be using the proxy server:

```
Defaults:username !requiretty
```

2. Under “root ALL=(ALL) ALL” add the following line for each account on an **ascp** client that will be using the proxy server:

```
username ALL = (ALL) NOPASSWD: /sbin/iptables-restore
```

Step 4 **Configure reverse proxy settings in aspera.conf.**

Locate and open the Aspera Proxy configuration file:

/opt/aspera/proxy/etc/aspera.conf

Look for the **<server>** tag that marks the beginning of the server section, or create it if necessary. Inside it, create an **<rproxy>** section where you will add one or more forwarding rules, marked by the **<rule>** tag. In particular, each rule will need at least one **<keyfile>** tag, which specifies the location of the SSH private keyfile in use, plus a valid **<host>** tag, which specifies the IP address and the SSH port that the server binds to.

For example, the **<rproxy>** description below specifies a configuration containing two valid forwarding rules. The first specifies that requests destined for proxy instance 7.7.7.7 should be forwarded to internal server 10.0.0.10. The second specifies that requests destined for 7.7.8.0/24 should be forwarded to internal server 10.0.0.30.

Rule (1) demonstrates a squashed user approach. All transfers that specify the 7.7.7.7 destination will be forwarded to squash-user **xfer** on internal server 10.0.0.10, and once there, they will be owned by **xfer**.

Rule (2) demonstrates the individual-account approach. When **<squash_user>** is not defined, the proxy server will use the proxy user’s account to authenticate with the internal server. For example, if a client connects with the proxy server with the username “diane” and no squash-user is specified, the proxy server will continue to use “diane” to authenticate with the internal server. If a rule will be used by multiple users, insert the **$(user)** variable in the keyfile path.

The second rule also demonstrates the use of a routing prefix, 7.7.8.0/24, instead of a single IP address for the proxy server instance. This covers destination requests specified anywhere in the range of 7.7.8.0 – 7.7.8.255.

```
<server>
    <rproxy>
        <enabled>true</enabled>
        <rules>
            <rule host_ip="7.7.7.7">
                <host>10.0.0.10:22</host>
                <squash_user>xfer</squash_user>
                <keyfile>/opt/aspera/proxy/etc/ssh_keys/id_rsa</keyfile>
            </rule>
            <rule host_ip="7.7.8.0/24">
                <host>10.0.0.30:22</host>
                <keyfile>/home/$(user)/.ssh/id_rsa</keyfile>
            </rule>
        </rules>
    </rproxy>
</server>
```

Note that the **<rproxy>** section must have **<enabled>** set to **true** to turn on reverse proxy.

The following tags can be used in a reverse proxy configuration:

Tag	Description	Default Value
<rule>	Rule with no conditional attributes.	N/A
<rule host_domain="*hostname***">**	Host name of the proxy server.	(none)
<rule host_ip="*ipaddr***">**	IP address of the proxy server.	(none)
<rule host_domain="*hostname***" host_ip="***ipaddr***">**	Combined version of the above.	(none)
<enabled>	Turn reverse proxy on/off (true/false).	false
<log_level>	Log only debug message level 1, debug message level 2, and so on.	0
<log_directory>	Proxy server log file location. If no value is set, Proxy logs to **syslog**.	blank (null)
<proxy_port>	Proxy server port that receives UDP traffic.	33001
<host>	Internal destination IP address and SSH port. The default port, if unspecified, is 22.	blank (null)
<squash_user>	Squash account name used for authenticating with the internal server.	blank (null)
<keyfile>	Path and file of the SSH private key for authenticating with the internal server.	blank (null)
<src_port_filtering>	Turn reverse proxy source-port filtering on or off (true/false). Setting this to false loosens reverse proxy security and therefore should be used only when necessary. For details, see the cautionary note below.	true

CAUTION: With source-port filtering turned on (default), reverse proxy restricts client connections to only those UDP source ports specified internally by each transfer session. In cases where client-side firewalls change the specified source port in transit, this option must be turned off to allow the connection to be established. Disabling source-port filtering relaxes reverse proxy security and therefore should be used only when necessary.

One indication that source-port filtering may need to be disabled is when client connections fail with a timeout such as “Error establishing UDP connection (check UDP port and firewall)”. Aspera transfer logs on either the client or server side will also show "Client unable to connect to server (check UDP port and firewall)” or “Server unable to hear from client (check UDP port and firewall)”. If the same timeout errors still occur when source-port filtering is disabled, this generally indicates that traffic is being blocked at a firewall.

With version 3.1 or later of the **ascp** client, rules can also be created with the **host_domain** option. For example, requests targeting **faspex.asperasoft.com** could be mapped to the first rule block, while requests targeting **shares.asperasoft.com** could be mapped to the second rule block.

To display a handy listing all reverse-proxy configuration options, you can run the **asuserdata** command as follows:

```
$ /opt/aspera/proxy/bin/asuserdata -s
```

NOTE: The **asuserdata -s** command displays the *default* values for the server setup, *not* the currently set values.

After running this command, scroll down to the **<!-- Server Options Spec -->** section. Configuration options for the reverse proxy server are displayed in the **<rproxy>** subsection:

```
<!-- Server Options Spec -->
...
<rproxy>
  <enabled>false</enabled>                <!-- rproxy_enabled: true|false -->
  <log_level>0</log_level>                <!-- log level: integer -->
  <log_directory></log_directory>         <!-- log dir: if no value (default), Proxy logs to syslog -->
  <rules>
    <rule>
      <proxy_port>33001</proxy_port>      <!-- rule proxy port: integer -->
      <host></host>                       <!-- internal host: IP address -->
      <squash_user></squash_user>         <!-- squash-user: user account name -->
      <keyfile></keyfile>                 <!-- keyfile: path and file -->
    </rule>
  </rules>
</rproxy>
  ...
```

Configuring Internal Servers

As with any destination node for **ascp** transfers, the internal node should be running Enterprise Server or Connect Server.

1. Log into the internal server node as root. Create an account for each user who will not be using the squashed account. Also create an account for the squash-user, if it will be used.
2. For each user, including the squash-user, create the file **/home/*user*/.ssh/authorized_keys** and append to it the public key generated for that user when you ran **ssh-keygen** on the proxy server.

Firewall Considerations

Your Aspera transfer products require access through the ports you have designated for SSH and UDP, typically port 33001. If you cannot establish connections, review your local corporate firewall settings and ensure that restrictions on these ports are removed accordingly. For details, see *Appendix A: Configuring Firewalls*.

Transferring Files with Reverse Proxy

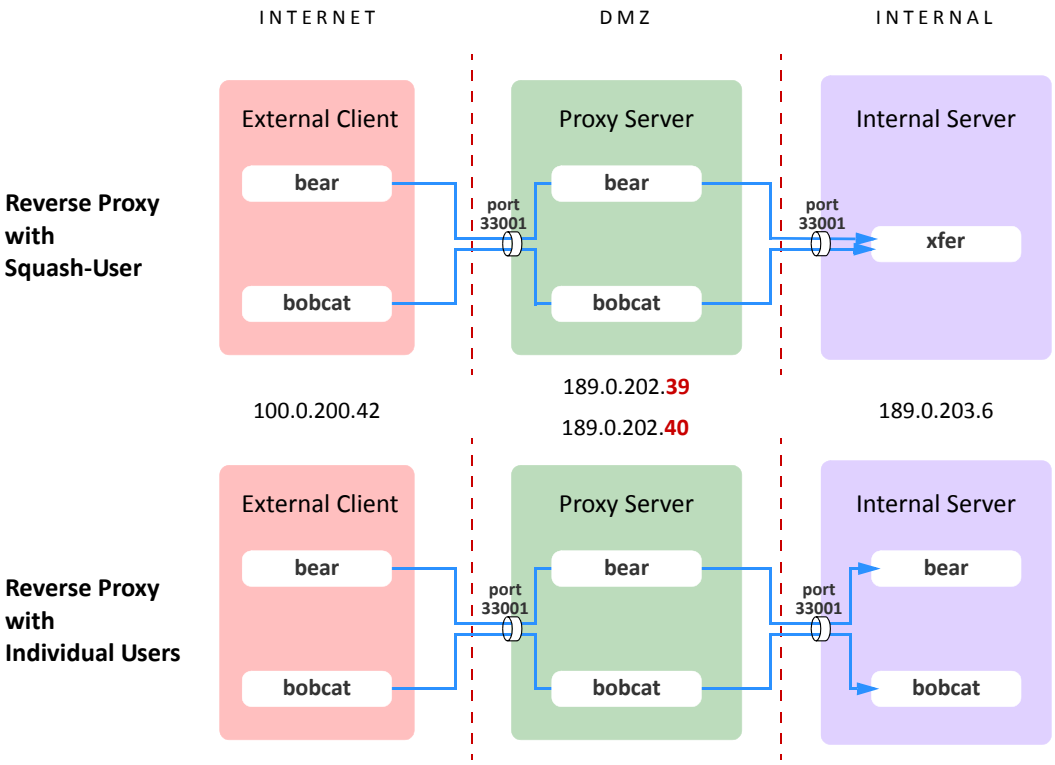
Once the configuration tasks have been completed for the proxy server, internal destination server, and external clients, file transfers from external users are completely transparent. To make transfers to the internal server, users need only specify the following:

- the IP address or domain name on the proxy server that corresponds to the internal destination
- the correct SSH port for the connection to the proxy server
- the target directory on the internal destination server

- any other parameters related to a transfer session (optional)

From the Command Line

The following configuration examples show the squash-user and individual-account approaches in the same system:



The reverse-proxy rules for each configuration are defined on the proxy server in **aspera.conf**:

```
<server>
  <rproxy>
    <enabled>true</enabled>
      <rules>
        <rule host_ip="189.0.202.39">
            <host>189.0.203.6:33001</host>
            <squash_user>xfer</squash_user>
            <keyfile>/opt/aspera/proxy/etc/ssh_keys/id_rsa</keyfile>
        </rule>
        <rule host_ip="189.0.202.40">
            <host>189.0.203.6:33001</host>
            <keyfile>/home/$(user)/.ssh/id_rsa</keyfile>
        </rule>
      </rules>
  </rproxy>
</server>
```

proxy instance for this rule

address and port of internal server

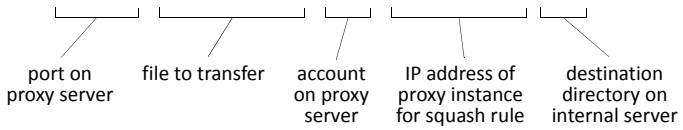
name of squash-user (**xfer**)

location of private key on proxy server

$(user) variable allows multiple users to specify this proxy instance

Users **bear** and **bobcat** have valid SSH key pairs and accounts on the proxy server. From the command line, **bear** runs the following **ascp** command specifying the proxy instance governed by the squash rule:

```
$ ascp -P 33001 testfile_bear bear@189.0.202.39:/tmp
```



Since the rule for proxy instance 189.0.202.**39** specifies a squash-user (**xfer**), the file belonging to **bear**, **bobcat**, or anyone using that proxy instance, will be owned by **xfer** when it arrives on the internal server.

The **-P 33001** flag specifies the port to use *on the proxy server* (not the port on the internal server, which is specified in the rule). The port must be specified on the command line if port 22 is disabled in **/etc/ssh/sshd_config**.

Users **bear** and **bobcat** have valid SSH key pairs and accounts on both the proxy server and the internal server. From the command line, **bobcat** runs the following **ascp** command specifying the proxy instance for the individual-user approach:

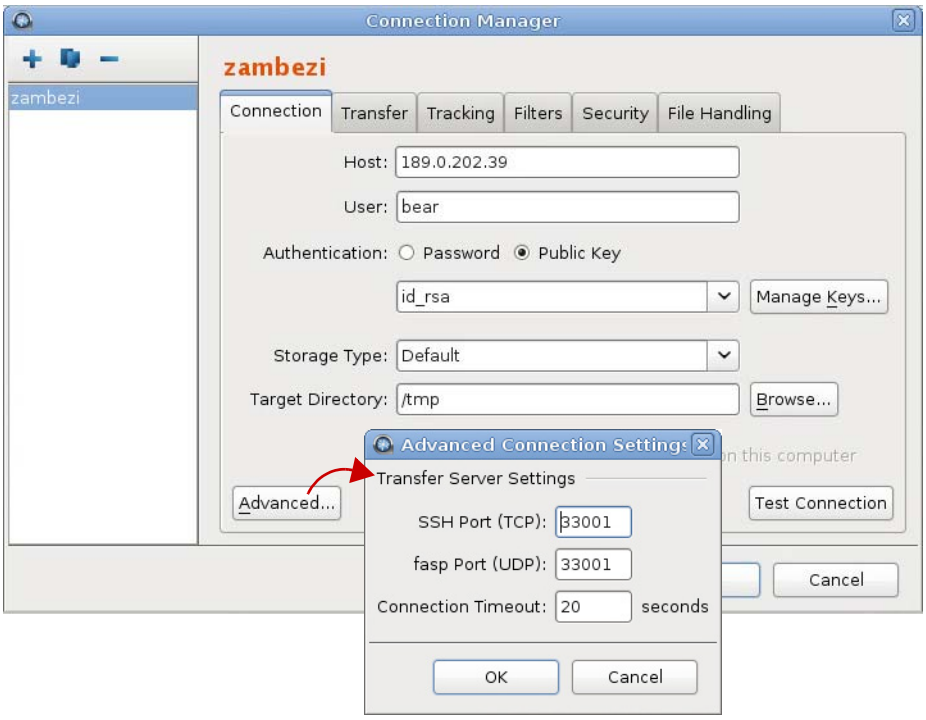
```
$ ascp -P 33001 testfile_bobcat bobcat@189.0.202.40:/tmp
```

Since the rule for proxy instance 189.0.202.**40** does not specify a squash-user, the file will still be owned by **bobcat** when it arrives on the internal server.

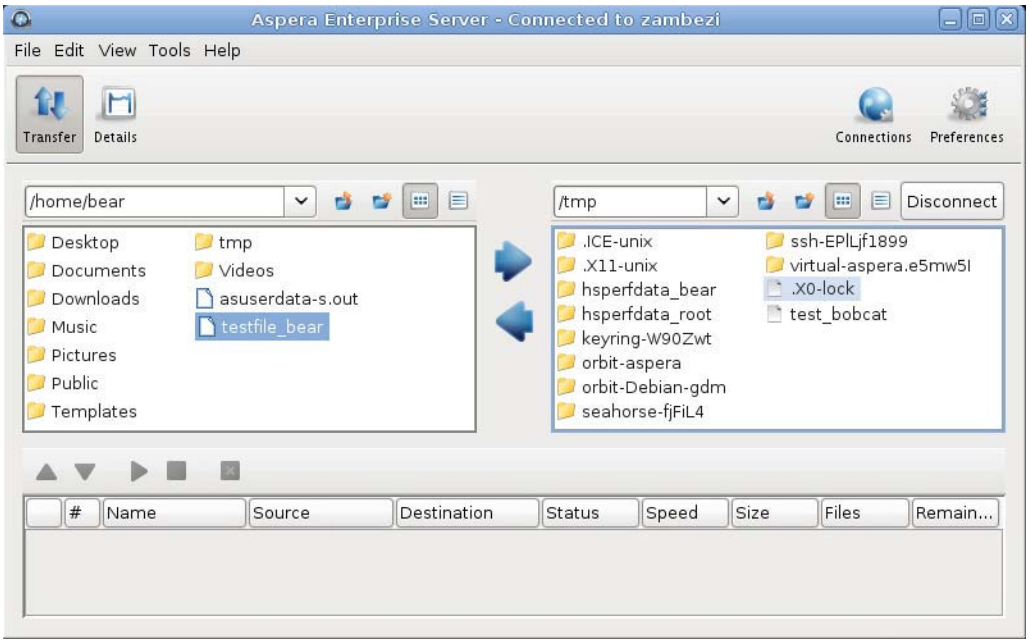
From the Enterprise Server GUI

All GUI-based Aspera transfer products can be used with Aspera Proxy, as well.

For example, user **bear** could also have made the above transfer with Enterprise Server. In the following display, **bear** has set up a connection called “zambezi” using the same parameters as above. The IP address of the proxy instance 189.0.202.39 (squash-user rule) is specified as the host. The filename for **bear**’s private SSH key is specified under Authentication/Public Key. The target directory on the internal server is specified as **/tmp**. The ports are specified as 33001 on the Advanced Connection Settings menu accessed from the **Advanced** button.



When **bear**'s connection to the proxy server is established, the **/tmp** target directory on the internal server is visible as in the right-hand panel in the display below, and ready for **bear** to make the transfer.



APPENDICES

Appendix A: Configuring Firewalls

Your Aspera transfer products require access through the ports listed below. If you cannot establish the connection, review your local corporate firewall settings and remove the port restrictions accordingly.

Aspera Enterprise Server

An Aspera server runs one SSH server on a configurable TCP port, 22 by standard default.

NOTE: Aspera strongly recommends running the SSH server on a non-default port to ensure that your server remains secure from SSH port scan attacks. Please see *Appendix B: Securing your SSH Server* for details on changing your SSH port.

Your firewall should be configured as follows:

- Allow inbound connections for SSH on a non-default, configurable TCP port. To ensure your server is secure, Aspera recommends allowing inbound connections for SSH on TCP/33001, and disallowing inbound connections on TCP/22. If you have a legacy customer base utilizing TCP/22, then you can allow inbound connections on both ports.
- Allow inbound connections for *fasp* transfers, which use UDP/33001 by default, although the server may also choose to run *fasp* transfers on another port.
- If you have a local firewall on your server (such as **iptables**), verify that it is not blocking your SSH and *fasp* transfer ports (e.g., TCP/UDP 33001).

The firewall on the server side must allow the open TCP port to reach the Aspera server. Note that no servers are listening on UDP ports. When a transfer is initiated by an Aspera client, the client opens an SSH session to the SSH server on the designated TCP port and negotiates the UDP port over which the data transfer will occur.

Aspera Client

Typically, consumer and business firewalls allow direct outbound connections from client computers on TCP and UDP. There is no configuration required for Aspera transfers in this case. In the special case of firewalls disallowing direct outbound connections, typically using proxy servers for Web browsing, the following configuration applies:

- Allow outbound connections from the Aspera client on the TCP port (TCP/33001, by default, when connecting to a Windows server, or on another non-default port for other server operating systems).
- Allow outbound connections from the Aspera client on the *fasp* UDP port (33001, by default).
- If you have a local firewall on your server (like **iptables**), verify that it is not blocking your SSH and *fasp* transfer ports (e.g. TCP/UDP 33001).

Appendix B: Securing your SSH Server

Keeping your data secure is critically important. Aspera strongly encourages you to take additional steps in setting up and configuring your SSH server so that it is protected against common attacks. Most automated robots will try to log into your SSH server on Port 22 as **root**, with various brute-force and dictionary combinations in order to gain access to your data. Furthermore, automated robots can put enormous loads on your server as they perform thousands of retries to break into your system. This topic addresses steps to take in securing your SSH server against potential threats, including changing the default port for SSH connections from TCP/22 to TCP/33001.

Why Change to TCP/33001?

It is well known that SSH servers listen for incoming connections on TCP port 22. As such, port 22 is subject to countless, unauthorized login attempts by hackers who are attempting to access unsecured servers. A highly effective deterrent is to simply turn off port 22 and run the service on a seemingly random port above 1024 (and up to 65535). To standardize the port for use in Aspera transfers, we recommend using TCP/33001.

NOTE: You need **root** access privileges to perform the steps below.

Step 1 **Locate and open the SSH configuration file on your system.**

Open your SSH configuration file with a text editor. You will find this file in the following system location:

/etc/ssh/sshd_config

Step 2 **Add new SSH port**

NOTE: Before changing the default port for SSH connections, please verify with your network administrators that TCP/33001 is open.

The OpenSSH suite included in the installer uses TCP/22 as the default port for SSH connections. Aspera recommends opening TCP/33001 and disabling TCP/22 to prevent security breaches of your SSH server.

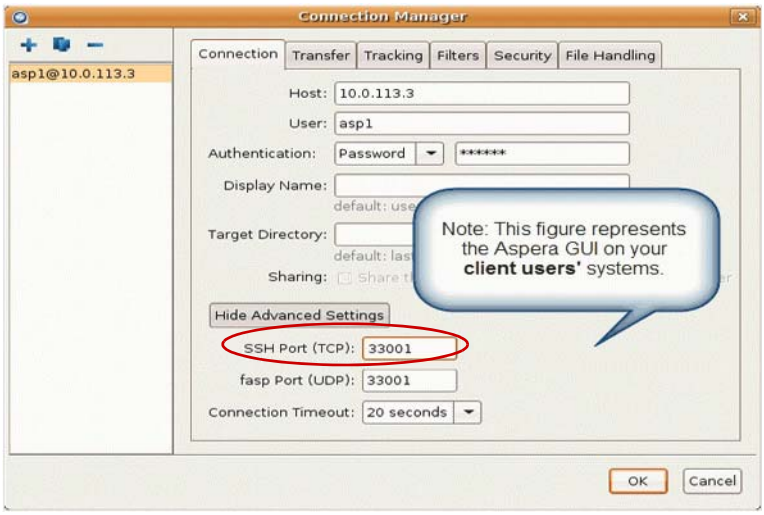
To enable TCP/33001 while your organization is migrating from TCP/22, open Port 33001 within your **sshd_config** file (where SSHD is listening on both ports). As demonstrated by this exercise, SSHD is capable of listening on multiple ports.

```
...
Port 22
Port 33001
...
```

Once your client users have been notified of the port change (from TCP/22 to TCP/33001), you can disable port 22 in your **sshd_config** file. To disable TCP/22 and use only TCP/33001, comment-out port 22 in your **sshd_config** file.

```
...
#Port 22
Port 33001
...
```

NOTE: Aspera recognizes that disabling the default SSH connection port (TCP/22) may affect your client users. When you change the port, ensure that you advise your users on configuring the new port number. Basic instructions for specifying the SSH port for *fasp* file transfers can be found below. To change the SSH port for Aspera Client, click **Connections** in the main window, and select the entry for your computer. Under the **Connection** tab, click **Show Advanced Settings** and enter the SSH port number in the **SSH Port (TCP)** field.



To make an impromptu connection to TCP/33001 during an **ascp** session, specify the SSH port (33001) with the **-P** (capital P) flag. Note that this command does not alter **ascp** or your SSH server's configuration.

```
$ ascp -P 33001 ...
```

Step 3 **Disable non-admin SSH tunneling.**

NOTE: The instructions below assume that OpenSSH 4.4 or newer is installed on your system. For OpenSSH 4.4 and newer versions, the "Match" directive allows some configuration options to be selectively overridden if specific criteria (based on user, group, hostname, and/or address) are met. If you are running an OpenSSH version older than 4.4, the "Match" directive will not be available and Aspera recommends that you update to the latest version.

In OpenSSH versions 4.4 and newer, disable SSH tunneling to avoid potential attacks, thereby allowing tunneling only from root users. To disable non-admin SSH tunneling, add the following lines at the end of the **sshd_config** file:

```
...
AllowTcpForwarding no
Match Group root
AllowTcpForwarding yes
```

Depending on your **sshd_config** file, you may have additional instances of **AllowTcpForwarding** that are set to the default **yes**. Please review your **sshd_config** file for other instances and disable as appropriate.

Note that disabling TCP forwarding does not improve security unless users are also denied shell access, as they can always install their own forwarders. Please review your user and file permissions, as well as refer to the instructions below on modifying shell access.

Step 4 **Update authentication methods.**

Public key authentication can prevent brute force SSH attacks if all password-based authentication methods are disabled. Thus, Aspera recommends disabling password authentication in the **sshd_config** file and enabling private/public key authentication. To do so, add or uncomment **PubkeyAuthentication yes** in the **sshd_config** file and comment out **PasswordAuthentication yes**.

```
...
PubkeyAuthentication yes
#PasswordAuthentication yes
PasswordAuthentication no
...
```

Step 5 **Disable root login.**

By default, OpenSSH allows root logins; however, disabling root access helps you to maintain a more secure server. Aspera recommends commenting out **PermitRootLogin yes** in the **sshd_config** file and adding **PermitRootLogin no**.

```
...
#PermitRootLogin yes
PermitRootLogin no
...
```

Administrators can then utilize the **su** or **sudo** commands if root privileges are needed.

Step 6 **Restart the SSH server to apply new settings.**

When you are finished updating your SSH server configuration, restart or reload the server to apply your new settings. Restarting or reloading your SSH server will not impact currently connected users. To restart or reload your SSH Server, use the following commands:

Red Hat Linux (restart):	`$ sudo service sshd restart`
Red Hat Linux (reload):	`$ sudo service sshd reload`
Debian Linux (restart):	`$ sudo /etc/init.d/ssh restart`
Debian Linux (reload):	`$ sudo /etc/init.d/ssh reload`

Step 7 **Review your user and file permissions.**

Permissions determine who can access certain files within your system, thereby making it a critical component of securing your server. By default, all user accounts are allowed to browse and read all files in the server.

To limit a user's access to a portion of the system, set the user account's shell to use the Aspera secured shell (**aspshell**) and set a document root for the user. The **aspshell** permits only the following operations:

- Run Aspera uploads and downloads to or from this computer.
- Establish connections in the application and browse, create, delete, rename, or list contents.

You configure **aspshell** behavior by editing **/opt/aspera/etc/aspera.conf**. The following template shows access options:

```
<file_system>
   <access>
      <paths>
         <path>
            <absolute>/sandbox/$(name)</absolute>      <!-- Absolute Path -->
            <read_allowed>true</read_allowed>          <!-- Read Allowed -->
            <write_allowed>true</write_allowed>        <!-- Write Allowed -->
            <dir_allowed>true</dir_allowed>            <!-- Browse Allowed -->
         </path>
      </paths>
   </access>
...
</file_system>
```

The following is a list of your Aspera product's docroot configuration options:

Tag	Description	Values	Default
<absolute>	The absolute path describes the area of the file system that is accessible by Aspera users. The default empty value gives users access to the entire file system.	file path	blank
<read_allowed>	Setting this to **true** allows users to transfer from the designated area of the file system as specified by the **<absolute>** value.	true false	blank
<write_allowed>	Setting this to **true** allows users to transfer to the designated area of the file system as specified by the **<absolute>** value.	true false	blank
<dir_allowed>	Setting this to **true** allows users to browse the directory.	true false	blank

Step 8 **Run the asp-check tool to check for potential user-security issues.**

The **asp-check** tool performs the following security checks:

- Searches for full-access users and reports how many exist on the system. Note that the existence of full-access users does not necessarily indicate that your system is vulnerable; however, it is being brought to the attention of the system administrator to ensure that the existence of full-access users is intentional.
- Searches for restricted users and potential misconfigurations, including:
  - incorrect login shell (i.e., one that is not restricted by **aspshell**)
  - SSH tunnel access (which can be used to work around the restricted shell)
  - docroot settings that allow users to access the home directory

NOTE: A docroot setting that allows access to the home directory does not necessarily indicate that your system is vulnerable; however, a user with this docroot can download or upload keys in **.ssh**, as well as upload **.login** scripts. These capabilities may be used to circumvent the intended, restricted nature of the user. Aspera highly recommends setting the docroot below the user's home folder (e.g., **/home/jane/data**) or in an alternate location (e.g., **/data**).

To run the **asp-check** tool, run the following command in a terminal window:

```
$ sudo /opt/aspera/proxy/bin/asp-check.sh
```

Search results appear in the terminal window, as shown in the example below. If potential issues are identified, review your users' settings before proceeding.

```
Users with full access: 22 (not considered insecure)
Restricted users: 0
Insecure users: 0
 - no restricted shell (aspshell): 0
 - docroot above home directory: 0
 - ssh tunneling enabled: 0
```

Step 9 **Review your logs periodically for attacks.**

Aspera recommends reviewing your SSH log periodically for signs of a potential attack. Locate and open your syslog; for example, **/var/log/auth.log** or **/var/log/secure**. Depending on your system configuration, syslog's path and file name may vary.

Look for invalid users in the log, especially a series of login attempts with common user names from the same address, usually in alphabetical order. For example:

```
...
Mar 10 18:48:02 sku sshd[1496]: Failed password for invalid user alex from 1.2.3.4
port 1585 ssh2
...
Mar 14 23:25:52 sku sshd[1496]: Failed password for invalid user alice from 1.2.3.4
port 1585 ssh2
...
```

If you have identified attacks:

- Double-check the SSH security settings described in this document.
- Report all attackers to your ISP's abuse email (e.g., abuse@your-isp).

Appendix C: Troubleshooting

Tracking connection status with proxy logs

The connection status for both forward and reverse proxy transfers is subject to regular logging in the system log file—**/var/log/messages** on Red Hat Linux and **/var/log/syslog** on Debian-based Linux. Root access is required for viewing the **syslog** file. The following is an example of a proxy transfer log entry triggered at the start of a transfer:

```
Dec  5 17:32:11 test1 ascp_rproxy[26250]: LOG Received connection request from 10.0.31.133
Dec  5 17:32:11 test1 ascp_rproxy[26250]: LOG Established SSH connection with server 10.0.30.6:22
Dec  5 17:32:11 test1 ascp_rproxy[26250]: LOG Setup UDP forwarding between 10.0.31.133:60953 and
10.0.30.6:33001
```

In the above:

10.0.31.133:60953 – client IP address and UDP port

10.0.30.6:22 – server IP and SSH port

10.0.30.6:33001 – IP and UDP port

The following is an example of a log entry when the connection is closed:

```
Dec  5 18:38:22 test1 ascp_rproxy[27238]: LOG Connection closed (EOF)
```

In the event of errors, individual error scenarios are logged separately.

To activate verbose debug logging, use the **<log_level>** tag in **aspera.conf** to set or increase the log-level value.

Error displays when trying to start node service

If you receive the following error when attempting to start the node service, check to see if **iptables** is installed on your machine:

```
ERR Failed to initialize proxy service
```

If **iptables** is not installed, issue the following command (based on your Linux distribution):

Red Hat Linux:

```
$ sudo yum install iptables
```

Debian-based Linux:

```
$ sudo apt-get install iptables
```

Using iptables to track forwarding rules

Proxy server administrators can also take advantage of the **iptables** tool to inspect the traffic forwarding rules that are in place.

For example, the following shows two DNAT rules, corresponding to two different **ascp** connections. The comment field of each rule contains the UUID of the **ascp** session. Note running the **iptables** command requires root privileges.

```
# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       udp  --  10.0.31.133          anywhere            udp spt:56393 dpt:33001 /*
3a9fd819-59c3-42bc-b0a2-d26304a1eb84 */ to:10.0.30.6:33001
DNAT       udp  --  10.0.31.133          anywhere            udp spt:44834 dpt:33001 /*
6dbcabeb-b71b-44a0-9e07-d0d9fccd5277 */ to:10.0.30.6:33001


Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere


Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

UDP Port and Firewall Timeout Errors

The following is a common timeout error:

```
Session Stop (Error: Client unable to connect to server -- check UDP port and
firewall.)
```

If you get this error, check the following:

1. Ensure that IP forwarding is enabled. IP forwarding must be enabled and is enabled automatically when Aspera Proxy is installed. To confirm, run the following command:

   ```
   $ cat /proc/sys/net/ipv4/ip_forward
   ```

   If the command returns 1, IP forwarding is enabled. If it returns 0, it is not. IP forwarding can be enabled manually by setting the **net.ipv4.ip_forward** line in **/etc/sysctl.conf** as follows:

   ```
   # Controls IP packet forwarding
   net.ipv4.ip_forward=1
   ```

   To activate changes to **/etc/sysctl.conf**, run the following:

   ```
   $ /sbin/sysctl -p /etc/sysctl.conf
   ```

2. If the error still occurs when IP forwarding is on, open your **aspera.conf** file and turn off source-port filtering as follows. By default, source-port filtering is enabled.

   ```
   ...
   <rproxy>
     <enabled>false</enabled>
     ...
     <rules>
       <rule>
         ...
         <src_port_filtering>false</src_port_filtering>
       </rule>
     </rules>
   </rproxy>
   ...
   ```

   If the same timeout errors still occur when source-port filtering is disabled, this generally indicates that traffic is being blocked at a firewall.

   For more information about source-port filtering, see CAUTION on p. 11.

DNAT Rules Left on the Proxy Server

On rare occasions, DNAT rules are left on the proxy server for sessions that have completed. To purge the rules, issue a stop and then a start to the proxy service:

```
$ /etc/init.d/asperaproxy stop
$ /etc/init.d/asperaproxy start
```