CARBONITE

Carbonite Availability for Linux

User's Guide

Notices

Carbonite Availability for Linux User's Guide Version 8.2.2, Thursday, January 3, 2019

If you need technical assistance, you can contact CustomerCare. All basic configurations outlined in the online documentation will be supported through CustomerCare. Assistance and support for advanced configurations may be referred to a Pre-Sales Systems Engineer or to Professional Services.

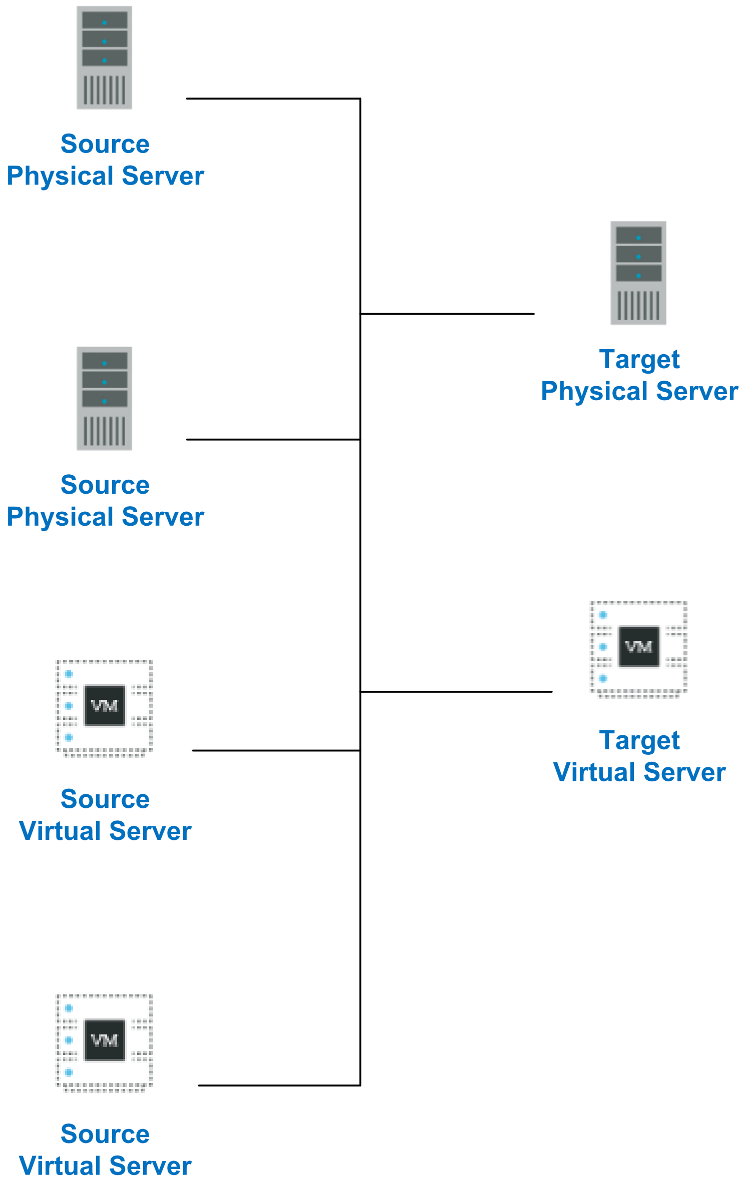
Man pages are installed and available on Carbonite Availability and Carbonite Migrate Linux servers. These documents are bound by the same Carbonite license agreement as the software installation.

Contents

Chapter 1 Carbonite Availability overview

Carbonite Availability ensures the availability of critical workloads. Using real-time replication and failover, you can protect data or entire servers, running on physical or virtual servers.

You identify what you want to protect on your production server, known as the source, and replicate that to a backup server, known as the target. The target server, on a local network or at a remote site, stores a replica copy of the data from the source. Carbonite Availability monitors any changes to the source and sends the changes to the replica copy stored on the target server. By replicating only the file changes rather than copying an entire file, Carbonite Availability allows you to more efficiently use resources.



Core operations

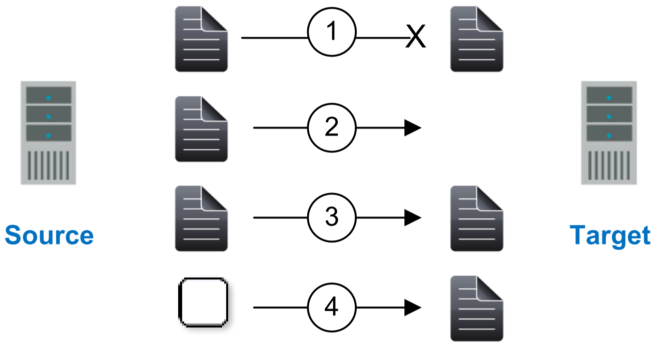
Carbonite Availability performs three basic types of operations.

- *Mirroring* on page 6—The initial copy or subsequent resynchronization of selected data
- *Replication* on page 7—The on-going capture of byte-level file changes
- *Failover* on page 8—The ability to stand-in for a server, in the event of a failure

Mirroring

Mirroring is the process of transmitting user-specified data from the source to the target so that an identical copy of data exists on the target. When Carbonite Availability initially performs mirroring, it copies all of the selected data, including file attributes and permissions. Mirroring creates a foundation upon which Carbonite Availability can efficiently update the target server by replicating only file changes.

If subsequent mirroring operations are necessary, Carbonite Availability can mirror specific files or blocks of changed data within files. By mirroring only files that have changed, network administrators can expedite the mirroring of data on the source and target servers. Mirroring has a defined end point when all of the selected files from the source have been transmitted to the target. When a mirror is complete, the target contains a copy of the source files at that point in time.

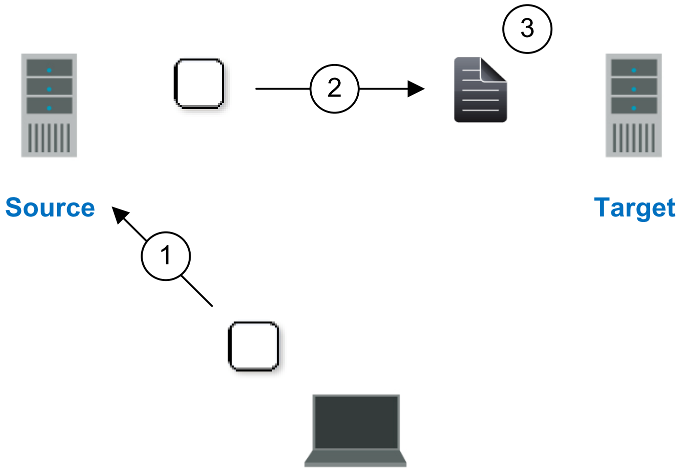


1. Identical files are not mirrored.
2. New files are mirrored.
3. Different files can be mirrored.
4. Checksums can calculate blocks of data to be mirrored.

Replication

Replication is the real-time transmission of file changes. Unlike other related technologies, which are based on a disk driver or a specific application, the Carbonite Availability replication process operates at the file system level and is able to track file changes independently from the file's related application. In terms of network resources and time, replicating changes is a more efficient method of maintaining a real-time copy of data than copying an entire file that has changed.

After a source and target have been connected through Carbonite Availability, file system changes from the user-defined data set are tracked. Carbonite Availability immediately transmits these file changes to the target server. This real-time replication keeps the data on the target up-to-date with the source and provides high availability and disaster recovery with minimal data loss. Unlike mirroring which is complete when all of the files have been transmitted to the target, replication continuously captures the changes as they are written to the source. Replication keeps the target up-to-date and synchronized with the source.



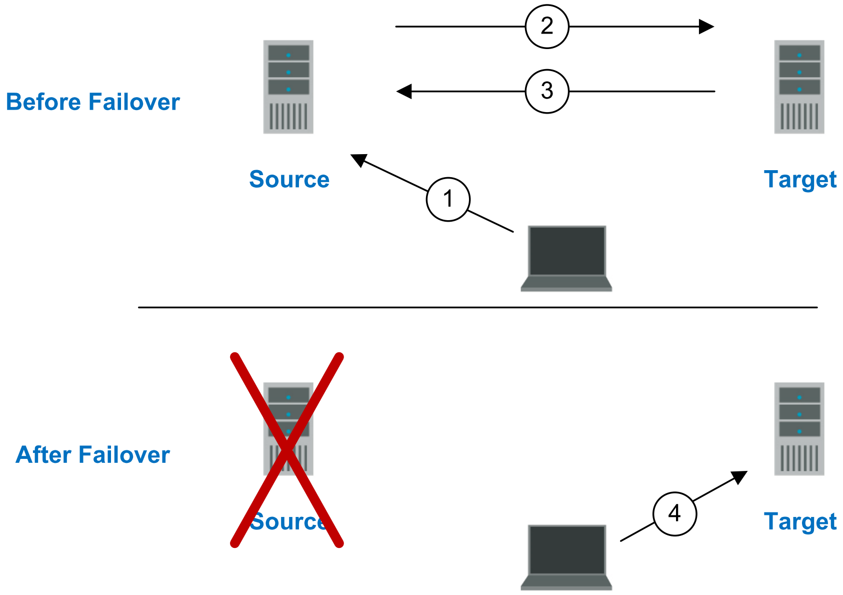
1. A user or application updates part of a file.
2. Only the changed portion of the file is replicated to the target.
3. An up-to-date copy of the file is maintained on the target.

Failover

Failover is the process in which a target stands in for a failed source. As a result, user and application requests that are directed to the failed source are routed to the target.

Carbonite Availability monitors the source status by tracking requests and responses exchanged between the source and target. When a monitored source does not respond to the target's requests, Carbonite Availability assumes that the server has failed. Carbonite Availability then prompts the network administrator to initiate failover, or, if configured, it occurs automatically. The failover target assumes the identity of the failed source, and user and application requests destined for the source server or its IP address(es) are routed to the target.

When partnered with the Carbonite Availability data replication capabilities, failover routes user and application requests with minimal disruption and little or no data loss.



1. User and application requests are sent to the source name or IP address.
2. Data on the source is mirrored and replicated to the target.
3. The target monitors the source for failure.
4. In the event the source fails, the target stands in for the source. User and application requests are still sent to the source name or IP address, which are now running on the target.

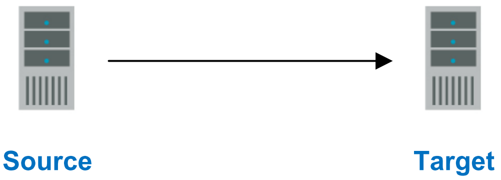
Supported configurations

Carbonite Availability is an exceptionally flexible product that can be used in a wide variety of network configurations. To implement Carbonite Availability effectively, it is important to understand the possible configuration options and their relative benefits. Carbonite Availability configurations can be used independently or in varying combinations.

Not all types of jobs support all of these configurations. See the requirements of each job type to determine which configurations are supported.

- *One to one, active/standby* on page 10
- *One to one, active/active* on page 11
- *Many to one* on page 12
- *One to many* on page 13
- *Chained* on page 14
- *Single server* on page 15

One to one, active/standby



Description

One target server, having no production activity, is dedicated to support one source server. The source is the only server actively replicating data.

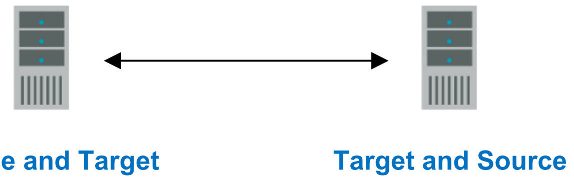
Applications

- This configuration is appropriate for offsite disaster recovery, failover, and critical data backup. This is especially appropriate for critical application servers.
- This is the easiest configuration to implement, support, and maintain.

Considerations

- This configuration requires the highest hardware cost because a target server is required for every source server.
- You must pause the target when backing up database files on the target.

One to one, active/active



Description

Each server acts as both a source and target actively replicating data to each other

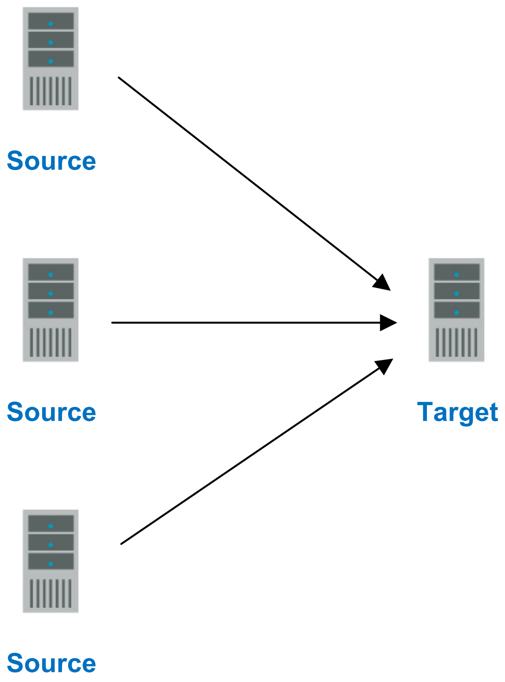
Applications

This configuration is appropriate for failover and critical data backup. This configuration is more cost-effective than the Active/Standby configuration because there is no need to buy a dedicated target server for each source. In this case, both servers can do full-time production work.

Considerations

- Coordination of the configuration of Carbonite Availability and other applications can be more complex than the one to one active/standby configuration.
- During replication, each server must continue to process its normal workload.
- Administrators must avoid selecting a target destination path that is included in the source’s protected data set. Any overlap will cause an infinite loop.
- To support the production activities of both servers during failover without reducing performance, each server should have sufficient disk space and processing resources.
- Failover and failback scripts must be implemented to avoid conflict with the existing production applications.
- You must pause the target when backing up database files on the target.

Many to one



Description

Many source servers are protected by one target server.

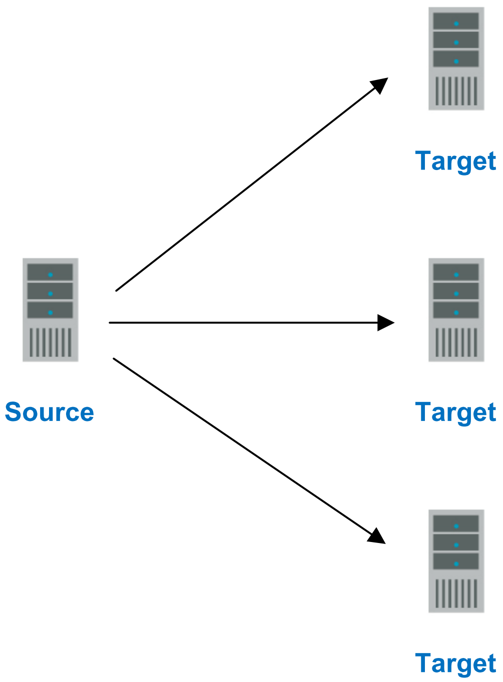
Applications

This configuration is appropriate for offsite disaster recovery. This is also an excellent choice for providing centralized tape backup because it spreads the cost of one target server among many source servers.

Considerations

- The target server must be carefully managed. It must have enough disk space and RAM to support replication from all of the source systems. The target must be able to accommodate traffic from all of the servers simultaneously.
- If using failover, scripts must be coordinated to ensure that, in the event that the target server stands in for a failed server, applications will not conflict.
- You must pause the target when backing up database files on the target.

One to many



Description

One source server sends data to multiple target servers. The target servers may or may not be accessible by one another.

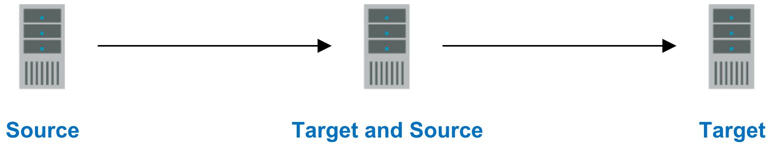
Applications

This configuration provides offsite disaster recovery, redundant backups, and data distribution. For example, this configuration can replicate all data to a local target server and separately replicate a subset of the mission-critical data to an offsite disaster recovery server.

Considerations

- Updates are transmitted multiple times across the network. If one of the target servers is on a WAN, the source server is burdened with WAN communications.
- You must pause the target when backing up database files on the target.
- If you failover to one of the targets, the other targets stop receiving updates.

Chained



Description

The source servers sends replicated data to a target server, which acts as a source server and sends data to a final target server, which is often offsite.

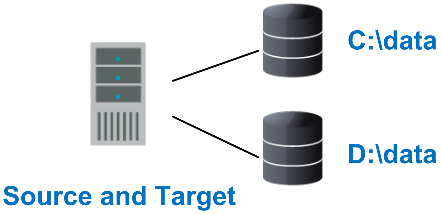
Applications

This is a convenient approach for integrating local high availability with offsite disaster recovery. This configuration moves the processing burden of WAN communications from the source server to the target/source server. After failover in a one to one, many to one, or one to many configuration, the data on the target is no longer protected. This configuration allows failover from the first source to the middle machine, with the third machine still protecting the data.

Considerations

- The target/source server could become a single point of failure for offsite data protection.
- You must pause the target when backing up database files on the target.

Single server



Description

Source and target components are loaded on the same server allowing data to be replicated from one location to another on the same volume or to a separate volume on the same server. These could be locally attached SCSI drives or Fibre Channel based SAN devices.

Applications

This configuration is useful upgrading storage hardware while leaving an application online. Once the data is mirrored, you can swap the drive in the disk manager. If the source and target copies of the data are located on different drives, this configuration supports high availability of the data in the event that the source hard drive fails.

Considerations

- This configuration does not provide high availability for the entire server.
- This configuration must be configured carefully so that an infinite loop is not created.
- This configuration should be limited to a single Carbonite Availability job.
- This configuration should be used sparingly. If possible, you should attach the target volumes to another server and use a one to one configuration.

Replication capabilities

Carbonite Availability replicates all file and directory data in the supported Linux file systems. Carbonite Availability does not replicate extended attributes (xattr) or items that are not stored on the file system, such as pseudo-file systems like /proc and /sys. In addition, note the following.

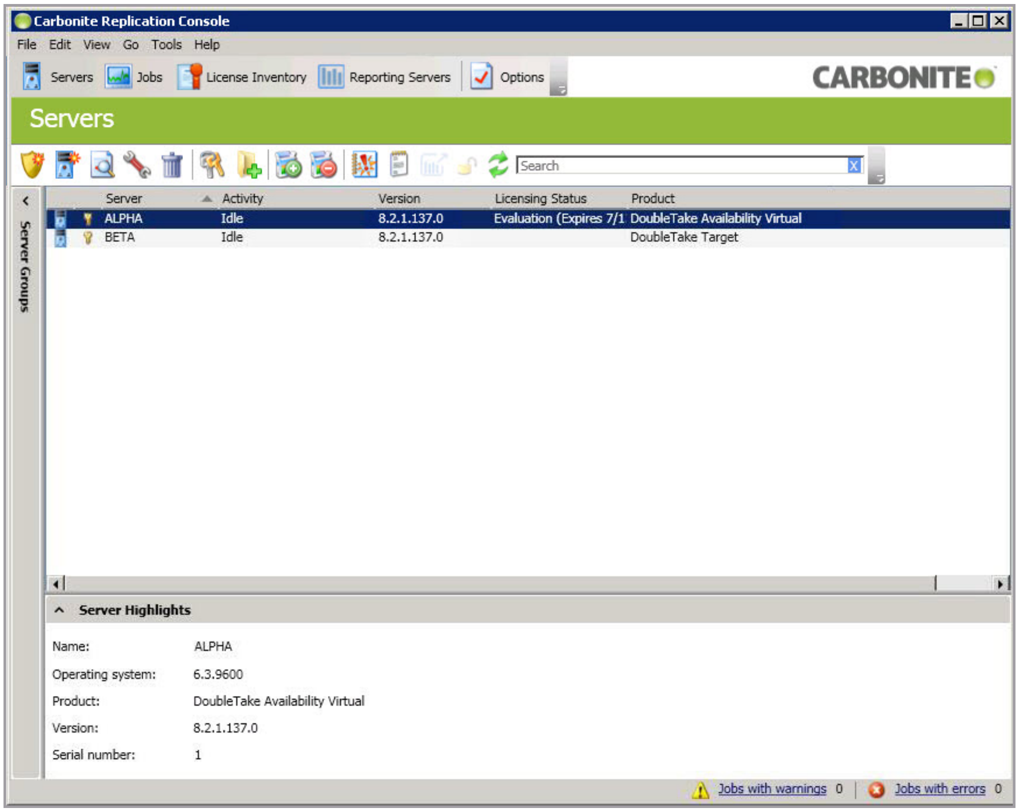
- Carbonite Availability is compatible with NFS and Samba services as long as they are mounted on top of Carbonite Availability. (The mount must be at the origination point, not a remote mounted point.) Additionally, NFS and Samba should be started after the Double-Take service.
- If you select data stored on a recursive mount point for replication, a mirror will never finish. Carbonite Availability does not check for data stored on recursive mount points.
- If any directory or file contained in your replication set specifically denies permission to the account running the Double-Take service, the attributes of the file on the target will not be updated because of the lack of access.
- Sparse files will become full size, zero filled files on the target.
- If you are using soft links, keep in mind the following.
    - If a soft link to a directory is part of a replication set rule's path above the entry point to the replication set data, that link will be created on the target as a regular directory if it must be created as part of the target path.
    - If a soft link exists in a replication set (or is moved into a replication set) and points to a file or directory inside the replication set, Carbonite Availability will remap the path contained in that link based on the Carbonite Availability target path when the option RemapLink is set to the default value (1). If RemapLink is set to zero (0), the path contained in the link will retain its original mapping.
    - If a soft link exists in a replication set (or is moved into a replication set) and points to a file or directory outside the replication set, the path contained in that link will retain its original mapping and is not affected by the RemapLink option.
    - If a soft link is moved out of or deleted from a replication set on the source, that link will be deleted from the target.
    - If a soft link to a file is copied into a replication set on the source and the operating system copies the file that the link pointed to rather than the link itself, then Carbonite Availability replicates the file copied by the operating system to the target. If the operating system does not follow the link, only the link is copied.
    - If a soft link to a directory is copied into a replication set on the source and the operating system copies the directory and all of its contents that the link pointed to rather than the link itself, then Carbonite Availability replicates the directory and its contents copied by the operating system to the target. If the operating system does not follow the link, only the link is copied.
    - If any operating system commands, such as chmod or chown, is directed at a soft link on the source and the operating system redirects the action to the file or directory which the link references, then if the file or directory referenced by the link is in a replication set, the operation will be replicated for that file to the target.
    - The operating system redirects all writes to soft links to the file referenced by the link. Therefore, if the file referenced by the symbolic link is in a replication set, the write operation will be replicated to the target.

- If you are using hard links, keep in mind the following.
    - If a hard link exists (or is created) only inside the replication set on the source, having no locations outside the replication set, the linked file will be mirrored to the target for all locations and those locations will be linked if all link locations on the target exist on the same partition.
    - If a hard link crosses the boundaries of a replication set on the source, having locations both inside and outside the replication set, the linked file will be mirrored to the target for only those locations inside the replication set on the source, and those locations will be linked on the target if all link locations exist on the same partition.
    - If a hard link is created on the source linking a file outside the replication set to a location inside the replication set, the linked file will be created on the target in the location defined by the link inside the replication set and will be linked to any other locations for that file which exist inside the replication set.
    - If any hard link location is moved from outside the replication set into the replication set on the source, the link will not be replicated to the target even if other link locations already exist inside the replication set, but the linked file will be created on the target in the location defined by the link.
    - If any hard link location existing inside the replication set is moved within the replication set on the source, the move will be replicated to the target and the link will be maintained if the new link location does not cross partitions in the target path.
    - If any hard link location existing inside the replication set is moved out of the replication set, that file or linked location will be deleted on the target.
    - If a hard linked file is copied from any location inside or outside the replication set to a location inside the replication set on the source, the copy will be replicated to the target.
    - If a hard linked file has a location in the replication set and any of the operating system commands, such as chmod or chown, are directed at that file from a location inside the replication set, the modification to the file will be replicated to the target. Operations on hard links outside of the replication set are not replicated.
    - If a hard linked file has a location in the replication set and a write operation is directed at that file from inside the replication set, the write operation will be replicated to the target. Operations on hard links outside of the replication set are not replicated.
    - If any hard link location existing inside the replication set is deleted on the source, that file or linked location will be deleted from the target.

Chapter 2 Carbonite Replication Console

After you have installed the console, you can launch it by selecting **Carbonite**, **Replication**, **Carbonite Replication Console** from your **Programs**, **All Programs**, or **Apps**, depending on your operating system.

The Carbonite Replication Console is used to protect and monitor your servers and jobs. Each time you open the Carbonite Replication Console, you start at the **Servers** page which allows you to view, edit, add, remove, or manage the servers in your console. You can also create a new job from this page.



Server	Activity	Version	Licensing Status	Product
ALPHA	Idle	8.2.1.137.0	Evaluation (Expires 7/1	DoubleTake Availability Virtual
BETA	Idle	8.2.1.137.0		DoubleTake Target

Server Highlights

Name:	ALPHA
Operating system:	6.3.9600
Product:	DoubleTake Availability Virtual
Version:	8.2.1.137.0
Serial number:	1

At the bottom of the Carbonite Replication Console, you will see a status bar. At the right side, you will find links for **Jobs with warnings** and **Jobs with errors**. This lets you see quickly, no matter which page of the console you are on, if you have any jobs that need your attention. Select this link to go to the **Jobs** page, where the appropriate **Filter: Jobs with warnings** or **Filter: Jobs with errors** will automatically be applied.

The first time you start the console, you will see the getting started screen tips on the **Servers** page. These tips walk you through the basic steps of adding a server to your console, installing Carbonite Availability on that server, and creating a job on that server. If you do not want to see the tips, close them. If you want to reopen the tips after you have closed them, select **Help**, **Show Getting Started Tips**.

You can manually check for Carbonite Availability updates by selecting **Help**, **Check for Updates**.

- **Update available**—If there is an update available, click **Get Update**. The dialog box will close and your web browser will open to the Carbonite web site where you can download and install the update.
- **No update available**—If you are using the most recent console software, that will be indicated. Click **Close**.
- **No connection available**—If the console cannot contact the update server of if there is an error, the console will report that information. The console log contains a more detailed explanation of the error. Click **Check using Browser** if you want to open your browser to check for console software updates. You will need to use your browser if your Internet access is through a proxy server.

Carbonite Replication Console requirements

You must meet the following requirements for the Carbonite Replication Console.

- **Operating system**—The Carbonite Replication Console can be run from a Windows source or target. It can also be run from a physical or virtual machine running Windows 10, Windows 8, or Windows 7 Service Pack 1 or later.
- **Microsoft .NET Framework**—Microsoft .NET Framework version 4.5.1 is required.
- **Screen resolution**—For best results, use a 1024x768 or higher screen resolution.

The Carbonite Availability installation prohibits the console from being installed on Server Core. Because Windows 2012 allows you to switch back and forth between Server Core and a full installation, you may have the console files available on Server Core, if you installed Carbonite Availability while running in full operating system mode. In any case, you cannot run the Carbonite Replication Console on Server Core.

Console options

There are several options that you can set that are specific to the Carbonite Replication Console. To access these console options, select **Options** from the toolbar.

- **Monitoring**—This section is used to determine how the console monitors your Carbonite Availability servers.
    - **Monitoring interval**—Specifies how often, in seconds, the console refreshes the monitoring data. The servers will be polled at the specified interval for information to refresh the console.
    - **Automatic retry**—This option will have the console automatically retry server login credentials, after the specified retry interval, if the server login credentials are not accepted. Keep in mind the following caveats when using this option.
        - This is only for server credentials, not job credentials.
        - A set of credentials provided for or used by multiple servers will not be retried for the specified retry interval on any server if it fails on any of the servers using it.
        - Verify your environment's security policy when using this option. Check your policies for failed login lock outs and resets. For example, if your policy is to reset the failed login attempt count after 30 minutes, set this auto-retry option to the same or a slightly larger value as the 30 minute security policy to decrease the chance of a lockout.
        - Restarting the Carbonite Replication Console will automatically initiate an immediate login.
        - Entering new credentials will initiate an immediate login using the new credentials.
    - **Retry on this interval**—If you have enabled the automatic retry, specify the length of time, in minutes, to retry the login.
- **Server Communication**—This section is used to determine how the console communicates with your Carbonite Availability servers.
    - **Default port for XML web services protocol**—Specifies the port that the console will use when sending and receiving data to Carbonite Availability servers. By default, the port is 6325. Changes to the console port will not take effect until the console is restarted.
    - **Default port for legacy protocol**—If you are using an older Carbonite Availability version, you will need to use the legacy protocol port. This applies to Carbonite Availability versions 5.1 or earlier.
- **Diagnostics**—This section assists with console troubleshooting.
    - **Export Diagnostic Data**—This button creates a raw data file that can be used for debugging errors in the Carbonite Replication Console. Use this button as directed by technical support.
    - **View Log File**—This button opens the Carbonite Replication Console log file. Use this button as directed by technical support. You can also select **View**, **View Console Log File** to open the Carbonite Replication Console log file.
    - **View Data File**—This button opens the Carbonite Replication Console data file. Use this button as directed by technical support. You can also select **View**, **View Console Data File** to open the Carbonite Replication Console data file.
- **License Inventory**—This section controls if the console contains a license inventory. This

feature may not appear in your console if your service provider has restricted access to it.

  - **Enable license inventory—**This option allows you to use this console to manage the Carbonite Availability licenses assigned to your organization. When this option is enabled, the **License Inventory** page is also enabled.

- **Default Installation Options**—All of the fields under the **Default Installation Options** section are used by the push installation on the **Install** page. The values specified here will be the default options used for the push installation.

  - **Activate online after install completes**—Specify if you want to activate your Carbonite Availability licenses at the end of the installation. The activation requires Internet access from the console machine or the machine you are installing to. Activation will be attempted from the console machine first and if that fails, it wil be attempted from the machine you are installing to. If you choose not to have the installation activate your licenses, you will have to activate them through the console license inventory or the server's properties page.
  - **Location of install folders—**Specify the parent directory location where the installation files are located. The parent directory can be local on your console machine or a UNC path.
    - **Windows—**Specify the parent directory where the Windows installation file is located. The default location is where the Carbonite Replication Console is installed, which is \Program Files\Carbonite\Replication. The console will automatically use the \x64 subdirectory which is populated with the Windows installation files when you installed the console. If you want to use a different location, you must copy the \x64 folder and its installation file to the different parent directory that you specify.
    - **Linux—**Specify the parent directory where the Linux installation files are located. The default location is where the Carbonite Replication Console is installed, which is \Program Files\Carbonite\Replication. The console will automatically use the \Linux subdirectory, however that location will not be populated with the Linux installation files when you installed the console. You must copy the Linux .deb or .rpm files from your download to the \Linux subdirectory in your Carbonite Replication Console installation location. Make sure you only have a single version of Linux installation files. The push installation cannot determine which version to install if there are multiple versions in the \Linux subdirectory. If you want to use a different location, you must copy the \Linux folder and its installation files to the different parent directory that you specify.

- **Default Windows Installation Options**—All of the fields under the **Default Installation Options** section are used by the push installation on the **Install** page. The values specified here will be the default options used for the push installation.

  - **Temporary folder for installation package—**Specify a temporary location on the server where you are installing Carbonite Availability where the installation files will be copied and run.
  - **Installation folder**—Specify the location where you want to install Carbonite Availability on each server. This field is not used if you are upgrading an existing version of Carbonite Availability. In that case, the existing installation folder will be used.
  - **Queue folder—**Specify the location where you want to store the Carbonite Availability disk queue on each server.
  - **Amount of system memory to use**—Specify the maximum amount of memory, in MB, that can be used for Carbonite Availability processing.

    - **Minimum free disk space**—This is the minimum amount of disk space in the specified **Queue folder** that must be available at all times. This amount should be less than the amount of physical disk space minus the disk size specified for **Limit disk space for queue**.
    - **Do not use disk queue**—This option will disable disk queuing. When system memory has been exhausted, Carbonite Availability will automatically begin the auto-disconnect process.
    - **Unlimited disk queue**—Carbonite Availability will use an unlimited amount of disk space in the specified **Queue folder** for disk queuing, which will allow the queue usage to automatically expand whenever the available disk space expands. When the available disk space has been used, Carbonite Availability will automatically begin the auto-disconnect process.
    - **Limit disk space for queue**—This option will allow you to specify a fixed amount of disk space, in MB, in the specified **Queue folder** that can be used for Carbonite Availability disk queuing. When the disk space limit is reached, Carbonite Availability will automatically begin the auto-disconnect process.

> If the servers you are pushing to do not have a C drive, make sure you update the folder fields because the Carbonite Replication Console will not validate that the fields are set to a volume that does not exist and the installation will not start.

- **Default Linux Installation Options**—All of the fields under the **Default Installation Options** section are used by the push installation on the **Install** page. The values specified here will be the default options used for the push installation.
    - **Temporary folder for installation package**—Specify a temporary location on the server where you are installing Carbonite Availability where the installation files will be copied and run.

Chapter 3 Managing servers

To manage the servers in your console, select **Servers** from the toolbar. The **Servers** page is for server management and job creation.

- **Add and remove servers**—You can add servers to and remove servers from the console.
- **View and edit**—You can view server details and edit Carbonite Availability server properties.
- **Create job**—You can create a protection or migration job for a selected server.
- **Server organization**—You can organize the servers that are in your console into groups, allowing you to filter the servers you are viewing based on your organization.

Review the following sections to understand the information and controls available on the **Servers** page.

If you have uninstalled and reinstalled Carbonite Availability on a server, you may see the server twice on the **Servers** page because the reinstall assigns a new unique identifier to the server. One of the servers (the original version) will show with the red X icon. You can safely remove that server from the console.

Left pane

You can expand or collapse the left pane by clicking on the **Server Highlights** heading. This pane allows you to organize your servers into folders. The servers displayed in the top right pane will change depending on the server group folder selected in the left pane. Every server in your console session is displayed when the **All Servers** group is selected. If you have created and populated server groups under **My Servers**, then only the servers in the selected group will be displayed in the right pane.

Between the main toolbar and the left pane is a smaller toolbar. These toolbar options control the server groups in the left pane.

Create New Server Group

Creates a new server group below the selected group

Rename Server Group

Allows you to rename the selected server group

Delete Server Group

Deletes the selected server group. This will not delete the servers in the group, only the group itself.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Top right pane

The top pane displays high-level overview information about your servers. You can sort the data within a column in ascending and descending order. You can also move the columns to the left or right of each other to create your desired column order. The list below shows the columns in their default left to right order.

Column 1 (Blank)

The first blank column indicates the machine type.

Carbonite Availability source or target server which could be a physical server, virtual machine, or a cluster node

Carbonite Availability source or target server which is a Windows cluster

vCenter server

ESX server

Carbonite Availability Reporting Service server

Offline server which means the console cannot communicate with this machine.

Any server icon with a red circle with white X overlay is an error which means the console can communicate with the machine, but it cannot communicate with Carbonite Availability on it.

Column 2 (Blank)

The second blank column indicates the security level

Processing—The console is attempting to communicate with machine.

Administrator access—This level grants full control.

Monitor only access—This level grants monitoring privileges only.

No security access—This level does not allow monitoring or control.

Server

The name or IP address of the server. If you have specified a reserved IP address, it will be displayed in parenthesis.

Activity

There are many different **Activity** messages that keep you informed of the server activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the server details. See *Viewing server details* on page 37.

Version

The Carbonite Availability product version information, if any.

Licensing Status

The status of the license, if any, on the server. If your license is expired, any jobs using that server will be in an error state. If you have multiple licenses, the status will indicate the license that requires the soonest action. For example, if you have a Carbonite Migrate license that expires in two days and a Carbonite Availability license that must be activated within 10 days, the status will be for the Carbonite Migrate license.

Product

The Carbonite Availability products, if any, licensed for the server

Bottom right pane

The details displayed in the bottom pane provide additional information for the server highlighted in the top pane. You can expand or collapse the bottom pane by clicking on the **Server Highlights** heading.

Name

The name or IP address of the server.

Operating system

The operating system of the server. This field will not be displayed if the console cannot connect to Carbonite Availability on the server.

Product

The Carbonite Availability products, if any, licensed for the server

Version

The product version information, if any

Serial Number

The serial number associated with the Carbonite Availability license

Toolbar

The following options are available on the main toolbar of the **Servers** page. Some options are only available for a single selected server and others are available for multiple selected servers.

Create a New Job

The available job creation choices depend on the Carbonite Availability licenses applied to your server.

- **Protect**—If you are licensed for Carbonite Availability, use the **Protect** option to create a protection job for the selected server.
- **Migrate**—If you are licensed for Carbonite Migrate or certain Carbonite Availability licenses, use the **Migrate** option to create a migration job for the selected server.

Add Servers

Adds a new server. This button leaves the **Servers** page and opens the **Add Servers** page. See *Adding servers* on page 34.

View Server Details

Views detailed information about a server. This button leaves the **Servers** page and opens the **View Server Details** page. See *Viewing server details* on page 37.

Edit Server Properties

Edits the server's properties and options. This button leaves the **Servers** page and opens the **Edit Server Properties** page. See *Editing server properties* on page 39.

Remove Server

Removes the server from the console.

Provide Credentials

Changes the login credentials that the Carbonite Replication Console use to authenticate to a server. This button opens the **Provide Credentials** dialog box where you can specify the new account information. See *Providing server credentials* on page 36. You will remain on the **Servers** page after updating the server credentials.

Manage Group Assignments

Allows you to assign, move, and remove the selected server from specific server groups. This buttons opens the Manage Group Assignments dialog box where you can assign and unassign the server to specific server groups. The server will appear in server groups marked with a checkmark, and will not appear in groups without a checkmark. Servers assigned to a server group will automatically appear in parent server groups.

Install

Installs or upgrades Carbonite Availability on the selected server. This button opens the **Install** page where you can specify installation options.

Uninstall

Uninstalls Carbonite Availability on the selected server.

View Server Events

Views Windows application event messages for a server. This option is not available for Linux sources or appliances.

View Server Logs

Views the Carbonite Availability logs messages for a server. This button opens the **Logs** window. This separate window allows you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple servers. When the Carbonite Replication Console is closed, all logging windows will automatically close.

Launch Reporting

Launches the Reporting Service report viewer.

Activate Online

Activates licenses and applies the activation keys to servers in one step. You must have Internet access for this process. You will not be able to activate a license that has already been activated.

Refresh

Refreshes the status of the selected servers.

Search

Allows you to search the product or server name for items in the list that match the criteria you have entered.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Right-click menu

The following options are available on the right-click menu of the **Servers** page. Some options are only available for a single selected server and others are available for multiple selected servers.

Protect

If you are licensed for Carbonite Availability, use the **Protect** option to create a protection job for the selected server.

Migrate

If you are licensed for Carbonite Migrate or certain Carbonite Availability licenses, use the **Migrate** option to create a migration job for the selected server.

View Server Details

Views detailed information about a server. This button leaves the **Servers** page and opens the **View Server Details** page. See *Viewing server details* on page 37.

Edit Server Properties

Edits the server's properties and options. This button leaves the **Servers** page and opens the **Edit Server Properties** page. See *Editing server properties* on page 39.

Remove Server

Removes the server from the console.

Provide Credentials

Changes the login credentials that the Carbonite Replication Console use to authenticate to a server. This button opens the **Provide Credentials** dialog box where you can specify the new account information. See *Providing server credentials* on page 36. You will remain on the **Servers** page after updating the server credentials.

Manage Group Assignments

Allows you to assign, move, and remove the selected server from specific server groups. This buttons opens the Manage Group Assignments dialog box where you can assign and unassign the server to specific server groups. The server will appear in server groups marked with a checkmark, and will not appear in groups without a checkmark. Servers assigned to a server group will automatically appear in parent server groups.

Install

Installs or upgrades Carbonite Availability on the selected server. This button opens the **Install** page where you can specify installation options.

Uninstall

Uninstalls Carbonite Availability on the selected server.

Copy

Copies the information for the selected servers. You can then paste the server information as needed. Each server is pasted on a new line, with the server information being comma-separated.

Paste

Pastes a new-line separated list of servers into the console. Your copied list of servers must be entered on individual lines with only server names or IP addresses on each line.

View Server Events

Views Windows event messages for a server. This option is not available for Linux sources or appliances.

View Server Logs

Views the Carbonite Availability logs messages for a server. This button opens the **Logs** window. This separate window allows you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple servers. When the Carbonite Replication Console is closed, all logging windows will automatically close.

Launch Reporting

Launches the Reporting Service report viewer.

Activate Online

Activates licenses and applies the activation keys to servers in one step. You must have Internet access for this process. You will not be able to activate a license that has already been activated.

Gather Support Diagnostics

Executes the diagnostic DTInfo utility which collects configuration data for use when reporting problems to technical support. It gathers Carbonite Availability log files; Carbonite Availability and system settings; network configuration information such as

IP, WINS, and DNS addresses; and other data which may be necessary for technical support to troubleshoot issues. You will be prompted for a location to save the resulting file which is created with the information gathered. Because this utility is gathering several pieces of information, across the network to your console machine, it may take several minutes to complete the information gathering and sending the resulting file to the console machine.

View Replication Service Details

Views the replication service details for a server. This option is not applicable to Linux source servers or appliances.

Refresh

Refreshes the status of the selected servers.

Adding servers

The first time you start the console, the **Servers** page is empty. In order to protect and monitor your servers, you must insert your servers and/or appliances in the console.

Inserting servers manually

1. Click **Servers** from the main toolbar.
2. Click **Add servers** from the **Servers** page toolbar.
3. On the **Manual Entry** tab, specify the server information.
   - **Server**—This is the name or IP address of the server or appliance to be added to the console.

     If you enter the source server's fully-qualified domain name, the Carbonite Replication Console will resolve the entry to the server short name. If that short name resides in two different domains, this could result in name resolution issues. In this case, enter the IP address of the server.

     If you are using a NAT environment, make sure you add your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.

   - **User name**—Specify a local user that is a member of the **dtadmin** or **dtmon** security group on the server.
   - **Password**—Specify the password associated with the **User name** you entered.
   - **Domain**—If you are working in a domain environment, specify the **Domain**.
   - **Management Service port**—If you want to change the port used by the Double-Take Management Service, disable **Use default port** and specify the port number you want to use. This option is useful in a NAT environment where the console needs to be able to communicate with the server using a specific port number. Use the public or private port depending on where the console is running in relation to the server you are adding.
4. After you have specified the server or appliance information, click **Add**.
5. Repeat steps 3 and 4 for any other servers or appliances you want to add.
6. If you need to remove servers or appliances from the list of **Servers to be added**, highlight a server and click **Remove**. You can also remove all of them with the **Remove All** button.
7. When your list of **Servers to be added** is complete, click **OK**.

Importing and exporting servers from a server and group configuration file

You can share the console server and group configuration between machines that have the Carbonite Replication Console installed. The console server configuration includes the server group configuration, server name, server communications ports, and other internal processing information.

To export a server and group configuration file, select **File**, **Export Servers**. Specify a file name and click **Save**. After the configuration file is exported, you can import it to another console.

When you are importing a console server and group configuration file from another console, you will not lose or overwrite any servers that already exist in the console. For example, if you have server alpha in your console and you insert a server configuration file that contains servers alpha and beta, only the server beta will be inserted. Existing group names will not be merged, so you may see duplicate server groups that you will have to manually update as desired.

To import a server and group configuration file, select **File**, **Import Servers**. Locate the console configuration file saved from the other machine and click **Open**.

Providing server credentials

To update the security credentials used for a specific server, select **Provide Credentials** from the toolbar on the **Servers** page. When prompted, specify the **User name**, **Password**, and **Domain** of the account you want to use for this server. Click **OK** to save the changes.

Viewing server details

Highlight a server on the **Servers** page and click **View Server Details** from the toolbar. The **View Server Details** page allows you to view details about that particular server. The server details vary depending on the type of server or appliance you are viewing.

Server name

The name or IP address of the server. If you have specified a reserved IP address, it will be displayed in parenthesis.

Operating system

The server's operating system version

Roles

The role of this server in your Carbonite Availability environment. In some cases, a server can have more than one role.

- **Engine Role**—Source or target server
- **Reporting Service**—Reporting Service server

Status

There are many different **Status** messages that keep you informed of the server activity. Most of the status messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the server details.

Activity

There are many different **Activity** messages that keep you informed of the server activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the server details.

Connected via

The IP address and port the server is using for communcations. You will also see the Carbonite Availability protocol being used to communicate with server. The protocol will be XML web services protocol (for servers running Carbonite Availability version 5.2 or later) or Legacy protocol (for servers running version 5.1 or earlier).

Version

The product version information

Access

The security level granted to the specified user

User name

The user account used to access the server

Licensing

Licensing information for the server

Source jobs

A list of any jobs from this server. Double-clicking on a job in this list will automatically open the **View Job Details** page.

Target jobs

A list of any jobs to this server. Double-clicking on a job in this list will automatically open the **View Job Details** page.

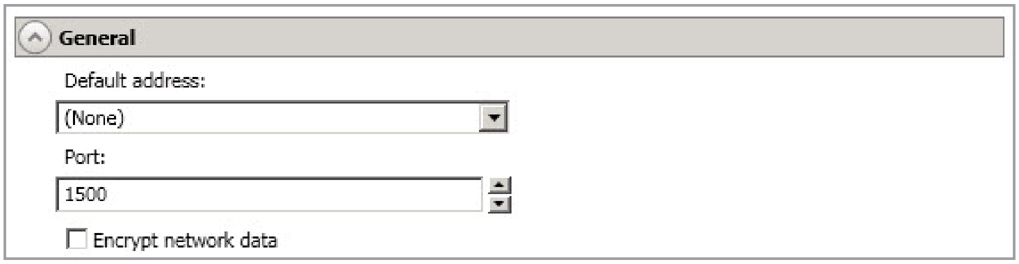
Editing server properties

Right-click a server on the **Servers** page and select **Edit server properties**. The **Edit Server Properties** page allows you to view and edit properties for that server. Click on a heading on the **Edit Server Properties** page to expand or collapse a section of properties.

- *General server properties* on page 40—Identifies the server and configures encryption
- *Server licensing* on page 41—Views, adds, and removes license keys
- *Server setup properties* on page 43—Indicates how the server will act on startup and shutdown
- *Carbonite Availability queue* on page 46—Configures the Carbonite Availability queues
- *Source server properties* on page 50—Configures the source server
- *Target server properties* on page 51—Configures the target server
- *Log file properties* on page 52—Configures log files
- *E-mail notification configuration* on page 54—Configures e-mail notification

General server properties

The general server properties identify the server and allow you to set encryption.



General

Default address:

(None)

Port:

1500

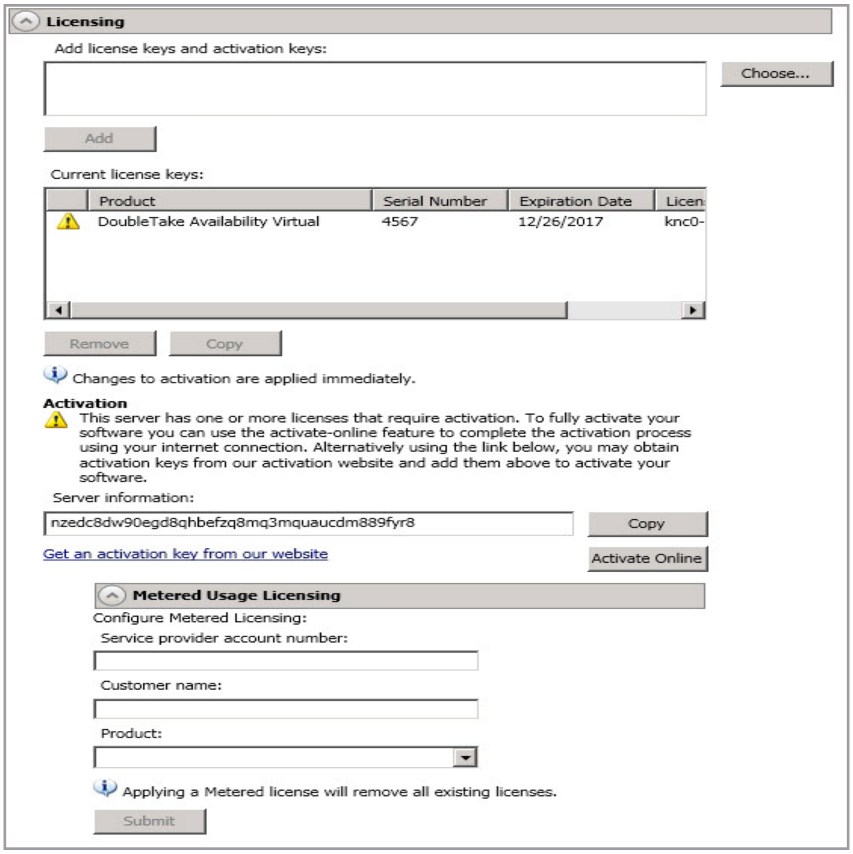
☐ Encrypt network data

- **Default address**—On a server with multiple NICs, you can specify which address Carbonite Availability traffic will use. It can also be used on servers with multiple IP addresses on a single NIC. If you change this setting, you must restart the Double-Take service for this change to take effect.
- **Port**—The server uses this port to send and receive commands and operations between Carbonite Availability servers. If you change the port, you must stop and restart the Double-Take service.
- **Encrypt network data**—Use this option to encrypt your data before it is sent from the source to the target. Both the source and target must be encryption capable (version 7.0.1 or later), however this option only needs to be enabled on the source or target in order to encrypt data. Keep in mind that all jobs from a source with this option enabled or to a target with this option enabled will have the same encryption setting. Changing this option will cause jobs to auto-reconnect and possibly remirror. The encryption method used is AES-256.

Server licensing

Licensing identifies your Carbonite Availability license keys.

The fields and buttons in the **Licensing** section will vary depending on your Carbonite Replication Console configuration and the type of license keys you are using.



Licensing

Add license keys and activation keys:

Current license keys:

	Product	Serial Number	Expiration Date	Licen
⚠	DoubleTake Availability Virtual	4567	12/26/2017	knc0-

Changes to activation are applied immediately.

Activation

⚠ This server has one or more licenses that require activation. To fully activate your software you can use the activate-online feature to complete the activation process using your internet connection. Alternatively using the link below, you may obtain activation keys from our activation website and add them above to activate your software.

Server information: nzedc8dw90egd8qhbefzq8mq3mquaucdm889fyr8

Get an activation key from our website

Metered Usage Licensing

Configure Metered Licensing:

Service provider account number:

Customer name:

Product:

Applying a Metered license will remove all existing licenses.

- **Add license keys and activation keys**—Your license key or activation key is a 24 character, alpha-numeric key. You can change your license key without reinstalling, if your license changes. To add a license key or activation key, type in the key or click **Choose from inventory** and select a key from your console's license inventory. Then click **Add**.

  The license inventory feature cannot be enabled if your service provider has restricted access to it.

- **Current license keys**—The server's current license key information is displayed. To remove a key, highlight it and click **Remove**. To copy a key, highlight it and click **Copy**. To replace a key, enter a new key and click **Add**. If you are replacing an unexpired key with the same version and serial number, you should not have to reactivate it and any existing jobs will continue uninterrupted. If you are replacing an unexpired key with a new version or new serial number or replacing an expired key, you will have to reactivate and remirror.

- **Activation**—If your license key needs to be activated, you will see an additional **Activation** section at the bottom of the **Licensing** section. To activate your key, use one of the following procedures.
  - **Activate online**—If you have Internet access, you can activate your license and apply the activated license to the server in one step by selecting **Activate Online**.

    You will not be able to activate a license that has already been activated.
  - **Obtain activation key online, then activate**—If you have Internet access, click the hyperlink in the **Activation** section to take you to the web so that you can submit your activation information. Complete and submit the activation form, and you will receive an e-mail with the activation key. Activate your server by entering the activation key in the **Add license keys and activations keys** field and clicking **Add**.
  - **Obtain activation key offline, then activate**—If you do not have Internet access, go to https://activate.doubletake.com from another machine that has Internet access. Complete and submit the activation form, and you will receive an e-mail with the activation key. Activate your server by entering the activation key in the **Add license keys and activations keys** field and clicking **Add**.

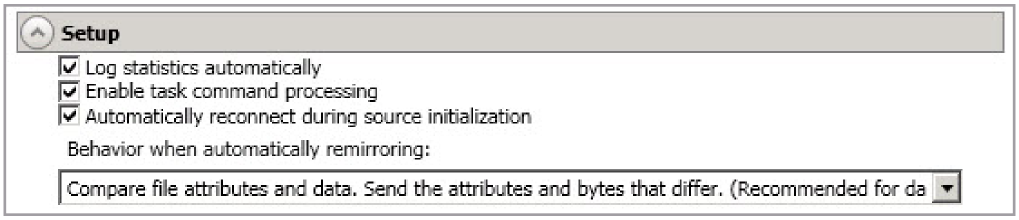
  The activation key is specific to this server. It cannot be used on any other server. If the activation key and server do not match, Carbonite Availability will not run.

  If your Carbonite Availability license keys needs to be activated, you will have 14 days to do so.

  If you need to rename a server that already has a Carbonite Availability license applied to it, you should deactivate that license before changing the server name. That includes rebuilding a server or changing the case (capitalization) of the server name (upper or lower case or any combination of case). If you have already rebuilt the server or changed the server name or case, you will have to perform a host-transfer to continue using that license.
- **Metered Usage Licensing**—If you are a service provider participating in the Metered Usage program, you can configure the metered usage license for your target servers here. If you are not in this program, you can skip this section. For the latest and complete details on Metered Usage, see the help link in the metered usage web portal.
  1. Specify your **Service provider account number**. The account number is displayed in the upper right corner of the metered usage portal.
  2. Specify the **Customer name**. Use the customer name configured on the Customers list in the metered usage portal.
  3. Select the appropriate **Product** that corresponds with the Carbonite Availability product being used.
  4. Click **Submit** to activate the metered usage license on the target.

Server setup properties

Server setup properties indicate how the server will act on startup and shutdown.



Setup

☑ Log statistics automatically

☑ Enable task command processing

☑ Automatically reconnect during source initialization

Behavior when automatically remirroring:

Compare file attributes and data. Send the attributes and bytes that differ. (Recommended for da

- **Log statistics automatically**—If enabled, Carbonite Availability statistics logging will start automatically when Carbonite Availability is started.
- **Enable task command processing**—Task command processing is a Carbonite Availability feature that allows you to insert and run tasks at various points during the replication of data. Because the tasks are user-defined, you can achieve a wide variety of goals with this feature. For example, you might insert a task to create a snapshot or run a backup on the target after a certain segment of data from the source has been applied on the target. This allows you to coordinate a point-in-time backup with real-time replication. Enable this option to enable task command processing, however to insert your tasks, you must use the Carbonite Availability scripting language. See the *Scripting Guide* for more information. If you disable this option on a source server, you can still submit tasks to be processed on a target, although task command processing must be enabled on the target.
- **Automatically reconnect during source initialization**—Disk queues are user configurable and can be extensive, but they are limited. If the amount of disk space specified for disk queuing is met, additional data would not be added to the queue and data would be lost. To avoid any data loss, Carbonite Availability will automatically disconnect jobs when necessary. If this option is enabled, Carbonite Availability will automatically reconnect any jobs that it automatically disconnected. These processes are called auto-disconnect and auto-reconnect and can happen in the following scenarios.
    - **Source server restart**—If your source server is restarted, Carbonite Availability will automatically reconnect any jobs that were previously connected. Then, if configured, Carbonite Availability will automatically remirror the data. This process is called auto-remirror. The remirror re-establishes the target baseline to ensure data integrity, so disabling auto-remirror is not advised.
    - **Exhausted queues on the source**—If disk queuing is exhausted on the source, Carbonite Availability will automatically start disconnecting jobs. This is called auto-disconnect. The transaction logs and system memory are flushed allowing Carbonite Availability to begin processing anew. The auto-reconnect process ensures that any jobs that were auto-disconnected are automatically reconnected. Then, if configured, Carbonite Availability will automatically remirror the data. This process is called auto-remirror. The remirror re-establishes the target baseline to ensure data integrity, so disabling auto-remirror is not advised.
    - **Exhausted queues on the target**—If disk queuing is exhausted on the target, the target instructs the source to pause. The source will automatically stop transmitting data to the target and will queue the data changes. When the target recovers, it will automatically tell the source to resume sending data. If the target does not recover by the time the source queues are exhausted, the source will auto-disconnect as described above. The

transaction logs and system memory from the source will be flushed then Carbonite Availability will auto-reconnect. If configured, Carbonite Availability will auto-remirror. The remirror re-establishes the target baseline to ensure data integrity, so disabling auto-remirror is not advised.

- **Queuing errors**—If there are errors during disk queuing on either the source or target, for example, Carbonite Availability cannot read from or write to the transaction log file, the data integrity cannot be guaranteed. To prevent any loss of data, the source will auto-disconnect and auto-reconnect. If configured, Carbonite Availability will auto-remirror. The remirror re-establishes the target baseline to ensure data integrity, so disabling auto-remirror is not advised.
- **Target server interruption**—If a target machine experiences an interruption (such as a cable or NIC failure), the source/target network connection is physically broken but both the source and target maintain the connection information. The Carbonite Availability source, not being able to communicate with the Carbonite Availability target, stops transmitting data to the target and queues the data changes, similar to the exhausted target queues described above. When the interruption is resolved and the physical source/target connection is reestablished, the source begins sending the queued data to the target. If the source/target connection is not reestablished by the time the source queues are exhausted, the source will auto-disconnect as described above.
- **Target service shutdown**—If the target service is stopped and restarted, there could have been data in the target queue when the service was stopped. To prevent any loss of data, the Double-Take service will attempt to persist to disk important target connection information (such as the source and target IP addresses for the connection, various target queue information, the last acknowledged operation, data in memory moved to disk, and so on) before the service is stopped. If Carbonite Availability is able to successfully persist this information, when the Double-Take service on the target is restarted, Carbonite Availability will pick up where it left off, without requiring an auto-disconnect, auto-reconnect, or auto-remirror. If Carbonite Availability cannot successfully persist this information prior to the restart (for example, a server crash or power failure where the target service cannot shutdown gracefully), the source will auto-reconnect when the target is available, and if configured, Carbonite Availability will auto-remirror. The remirror re-establishes the target baseline to ensure data integrity, so disabling auto-remirror is not advised.

If you are experiencing frequent auto-disconnects, you may want to increase the amount of disk space on the volume where the Carbonite Availability queue is located or move the disk queue to a larger volume.

If you have manually changed data on the target, for example if you were testing data on the target, Carbonite Availability is unaware of the target data changes. You must manually remirror your data from the source to the target, overwriting the target data changes that you caused, to ensure data integrity between your source and target.

- **Behavior when automatically remirroring**—Specify how Carbonite Availability will perform the mirror when it is automatically remirroring.

If you are using files and folders, full server (Linux) to ESX, or files and folders migration job and are using a database application or are protecting a domain controller, do not use the compare file attributes only options unless you know for certain that you need it. With database applications and because domain controllers store their data in a database, it is critical that all files, not just some of the files, are mirrored. In this case, you should compare both the attributes and the data.

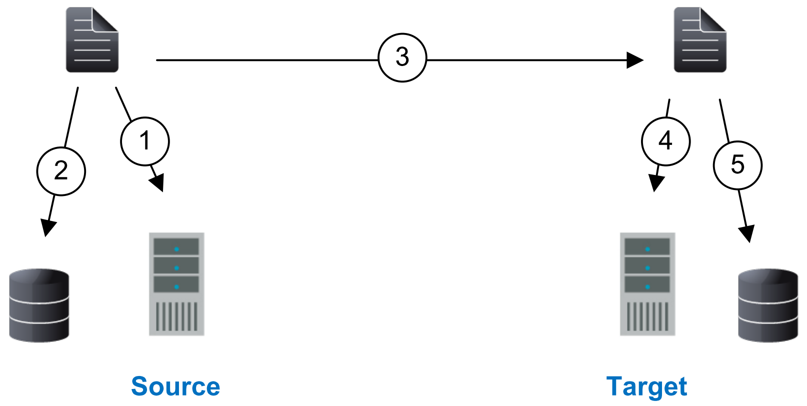
- **Do not compare files. Send the entire file.**—Carbonite Availability will not perform any comparisons between the files on the source and target. All files will be mirrored to the target, sending the entire file.
- **Compare file attributes. Send the entire file.**—Carbonite Availability will compare file attributes and will mirror those files that have different attributes, sending the entire file.
- **Compare file attributes. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and will mirror only the attributes and bytes that are different.
- **Compare file attributes and data. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and the file data and will mirror only the attributes and bytes that are different.

Carbonite Availability queue

During the Carbonite Availability installation, you identified the amount of disk space that can be used for Carbonite Availability queuing. Queuing to disk allows Carbonite Availability to accommodate high volume processing that might otherwise exhaust system memory. For example, on the source, this may occur if the data is changing faster than it can be transmitted to the target, or on the target, a locked file might cause processing to back up.

Carbonite Availability Queuing Diagram

The following diagram will help you understand how queuing works. Each numbered step is described after the diagram.



1. If data cannot immediately be transmitted to the target, it is stored in system memory. You can configure how much system memory you want Carbonite Availability to use for all of its processing.
2. When the allocated amount of system memory is full, new changed data bypasses the full system memory and is queued directly to disk. Data queued to disk is written to a transaction log. Each transaction log can store 5 MB worth of data. Once the log file limit has been reached, a new transaction log is created. The logs can be distinguished by the file name which includes the target IP address, the Carbonite Availability port, the connection ID, and an incrementing sequence number.

   You may notice transaction log files that are not the defined size limit. This is because data operations are not split. For example, if a transaction log has 10 KB left until the limit and the next operation to be applied to that file is greater than 10 KB, a new transaction log file will be created to store that next operation. Also, if one operation is larger than the defined size limit, the entire operation will be written to one transaction log.

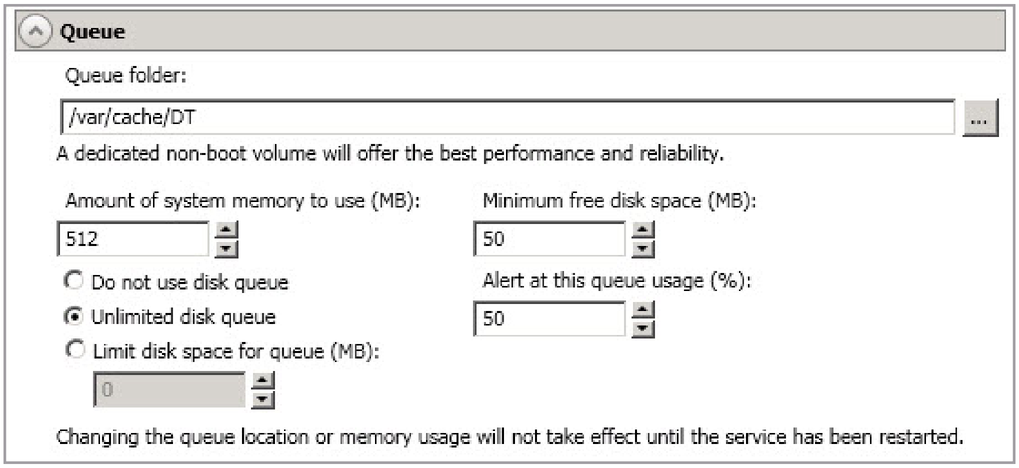
3. When system memory is full, the most recent changed data is added to the disk queue, as described in step 2. This means that system memory contains the oldest data. Therefore, when

data is transmitted to the target, Carbonite Availability pulls the data from system memory and sends it. This ensures that the data is transmitted to the target in the same order it was changed on the source. Carbonite Availability automatically reads operations from the oldest transaction log file into system memory. As a transaction log is depleted, it is deleted. When all of the transaction log files are deleted, data is again written directly to system memory (step 1).

4. To ensure the integrity of the data on the target, the information must be applied in the same order as it was on the source. If there are any delays in processing, for example because of a locked file, a similar queuing process occurs on the target. Data that cannot immediately be applied is stored in system memory.
5. When the allocated amount of system memory on the target is full, new incoming data bypasses the full system memory and is queued directly to disk. Data queued to disk is written to a transaction log. On the target, the transaction logs are identified with the source IP address, the Carbonite Availability port, the connection ID, and an incrementing sequence number.

   Like the source, system memory on the target contains the oldest data so when data is applied to the target, Carbonite Availability pulls the data from system memory. Carbonite Availability automatically moves operations from the oldest transaction log file to system memory. As a transaction log is depleted, it is deleted. When all of the transaction log files are deleted, data is again written directly to system memory (step 4).

The following memory and queue options are available for each Carbonite Availability server.



Queue

Queue folder: /var/cache/DT

A dedicated non-boot volume will offer the best performance and reliability.

Amount of system memory to use (MB): 512

Minimum free disk space (MB): 50

○ Do not use disk queue

● Unlimited disk queue

○ Limit disk space for queue (MB): 0

Alert at this queue usage (%): 50

Changing the queue location or memory usage will not take effect until the service has been restarted.

- **Queue folder**—This is the location where the disk queue will be stored. Any changes made to the queue location will not take effect until the Double-Take service has been restarted on the server.

  When selecting the queue location, keep in mind the following caveats.

  - Select a dedicated, non-boot volume.
  - Do not select the same physical or logical volume as the data being replicated.
  - Do not select the root of a volume.

  Although the read/write ratio on queue files will be 1:1, optimizing the disk for write activity will benefit performance because the writes will typically be occurring when the server is under a high load, and more reads will be occurring after the load is reduced. Accordingly, use a standalone disk, mirrored (RAID 1) or non-parity striped (RAID 0) RAID set, and allocate more I/O adapter cache memory to writes for best performance. A RAID 5 array will not perform as well as a mirrored or non-parity striped set because writing to a RAID 5 array incurs the overhead of generating and writing parity data. RAID 5 write performance can be up to 50% less than the write performance of a single disk, depending on the adapter and disk.

Scanning the Carbonite Availability queue files for viruses can cause unexpected results. If anti-virus software detects a virus in a queue file and deletes or moves it, data integrity on the target cannot be guaranteed. As long as you have your anti-virus software configured to protect the actual production data, the anti-virus software can clean, delete, or move an infected file and the clean, delete, or move will be replicated to the target. This will keep the target from becoming infected and will not impact the Carbonite Availability queues.

- **Amount of system memory to use**—This is the maximum amount of Windows system memory, in MB, that Carbonite Availability will use. When this limit is reached, queuing to disk will be triggered. The minimum amount of system memory is 512 MB. The maximum amount is dependent on the server hardware and operating system. If you set this value lower, Carbonite Availability will use less system memory, but you will queue to disk sooner which may impact system performance. If you set it higher, Carbonite Availability will maximize system performance by not queuing to disk as soon, but the system may have to swap the memory to disk if the system memory is not available.

  Since the source is typically running a production application, it is important that the amount of memory Carbonite Availability and the other applications use does not exceed the amount of RAM in the system. If the applications are configured to use more memory than there is RAM, the system will begin to swap pages of memory to disk and the system performance will degrade. For example, by default an application may be configured to use all of the available system memory when needed, and this may happen during high-load operations. These high-load operations cause Carbonite Availability to need memory to queue the data being changed by the application. In this case, you would need to configure the applications so that they collectively do not exceed the amount of RAM on the server. Perhaps on a server with 4 GB of RAM running the application and Carbonite Availability, you might configure the application to use 1 GB and Carbonite Availability to use 1 GB, leaving 2 GB for the operating system and other applications on the system. Many server applications default to using all available system memory, so it is important to check and configure applications appropriately, particularly on high-capacity servers.

  Any changes to the memory usage will not take effect until the Double-Take service has been restarted on the server.
- **Do not use disk queue**—This option will disable disk queuing. When system memory has been exhausted, Carbonite Availability will automatically begin the auto-disconnect process.
- **Unlimited disk queue**—Carbonite Availability will use an unlimited amount of disk space in the specified **Queue folder** for disk queuing, which will allow the queue usage to automatically expand whenever the available disk space expands. When the available disk space has been used, Carbonite Availability will automatically begin the auto-disconnect process.
- **Limit disk space for queue**—This option will allow you to specify a fixed amount of disk space, in MB, in the specified **Queue folder** that can be used for Carbonite Availability disk queuing. When the disk space limit is reached, Carbonite Availability will automatically begin the auto-disconnect process.
- **Minimum free disk space**—This is the minimum amount of disk space in the specified **Queue folder** that must be available at all times. This amount should be less than the amount of physical

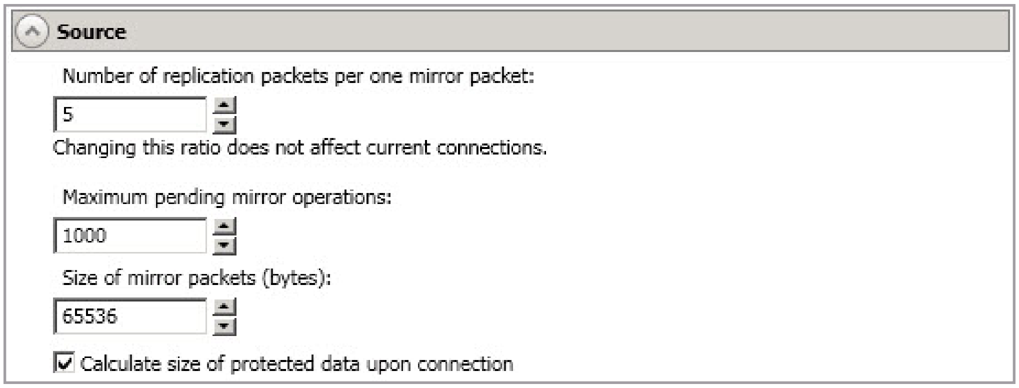
disk space minus the disk size specified for **Limit disk space for queue**.

The **Limit disk space for queue** and **Minimum free disk space** settings work in conjunction with each other. For example, assume your queue is stored on a 10 GB disk with the **Limit disk space for queue** set to 10 GB and the **Minimum free disk space** set to 500 MB. If another program uses 5 GB, Carbonite Availability will only be able to use 4.5 GB so that 500 MB remains free.

- **Alert at this queue usage**—This is the percentage of the disk queue that must be in use to trigger an alert message. By default, the alert will be generated when the queue reaches 50%.

Source server properties

These properties are specific to the source server role.



Source

Number of replication packets per one mirror packet:

5

Changing this ratio does not affect current connections.

Maximum pending mirror operations:

1000

Size of mirror packets (bytes):

65536

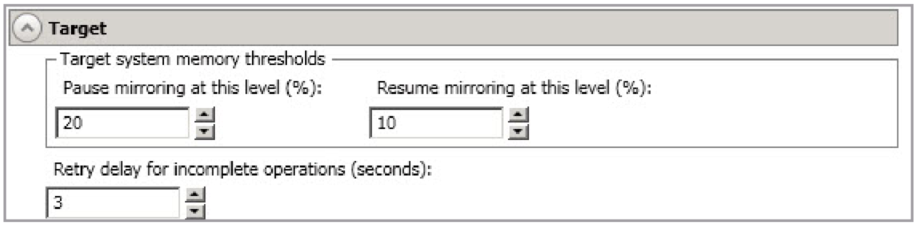
☑ Calculate size of protected data upon connection

- **Number of replication packets per one mirror packet**—You can specify the ratio of replication packets to mirror packets that are placed in the source queue. The default value (5) allows Carbonite Availability to dynamically change the ratio as needed based on the amount of replication data in queue. If you set a specific value other than the default (other than 5), the specified value will be used. Changes to this setting will take effect for future jobs. Existing jobs will have to be stopped and restarted to pick up the new ratio.
- **Maximum pending mirror operations**—This option is the maximum number of mirror operations that are queued on the source. The default setting is 1000. If, during mirroring, the mirror queued statistic regularly shows low numbers, for example, less than 50, this value can be increased to allow Carbonite Availability to queue more data for transfer.
- **Size of mirror packets**—This option determines the size of the mirror packets, in bytes, that Carbonite Availability transmits. The default setting is 65536 bytes. You may want to consider increasing this value in a high latency environment (greater than 100 ms response times), or if your data set contains mainly larger files, like databases.
- **Calculate size of protected data upon connection**—Specify if you want Carbonite Availability to determine the mirroring percentage calculation based on the amount of data being protected. If you enable this option, the calculation will begin when mirroring begins. For the initial mirror, the percentage will display after the calculation is complete, adjusting to the amount of the mirror that has completed during the time it took to complete the calculation. Subsequent mirrors will initially use the last calculated size and display an approximate percentage. Once the calculation is complete, the percentage will automatically adjust down or up to indicate the amount that has been completed. Disabling calculation will result in the mirror status not showing the percentage complete or the number of bytes remaining to be mirrored.

  The calculated amount of protected data may be slightly off if your data set contains compressed or sparse files.

Target server properties

These properties are specific to the target server role.



Target

Target system memory thresholds

Pause mirroring at this level (%): 20

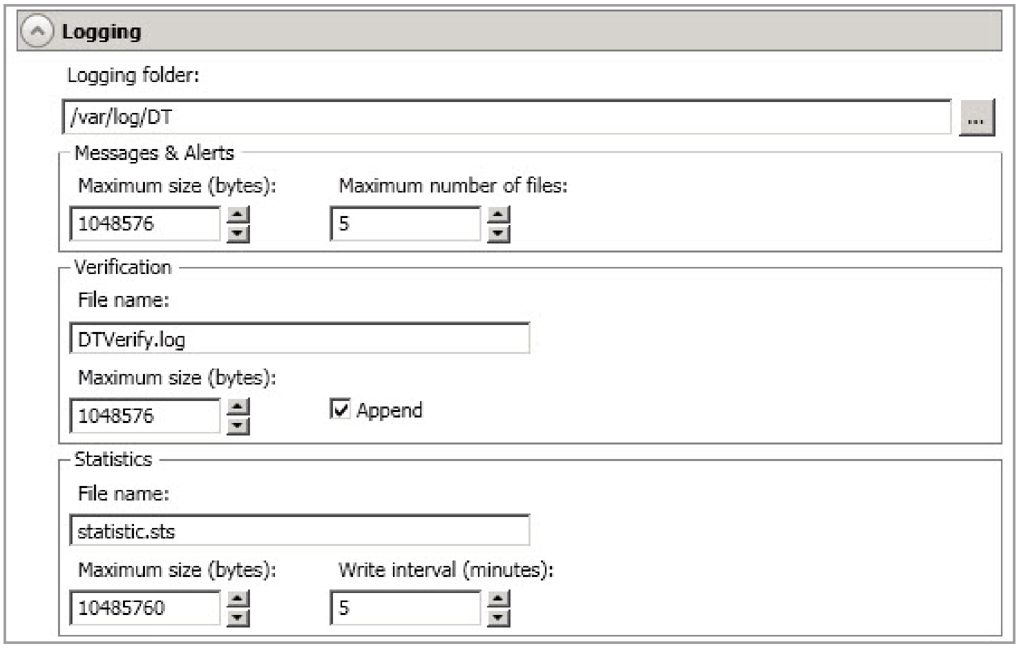
Resume mirroring at this level (%): 10

Retry delay for incomplete operations (seconds): 3

- **Pause mirroring at this level**—You can specify the maximum percentage of Windows system memory that can contain mirror data before the target signals the source to pause the sending of mirror operations. The default setting is 20.
- **Resume mirroring at this level**—You can specify the minimum percentage of Windows system memory that can contain mirror data before the target signals the source to resume the sending of mirror operations. The default setting is 15. You cannot set the resume value higher than the pause value.
- **Retry delay for incomplete operations—**This option specifies the amount of time, in seconds, before retrying a failed operation on the target. The default setting is 3.

Log file properties

These settings allow you to specify your log file configuration.



Logging

Logging folder: /var/log/DT

Messages & Alerts

Maximum size (bytes): 1048576

Maximum number of files: 5

Verification

File name: DTVerify.log

Maximum size (bytes): 1048576

☑ Append

Statistics

File name: statistic.sts

Maximum size (bytes): 10485760

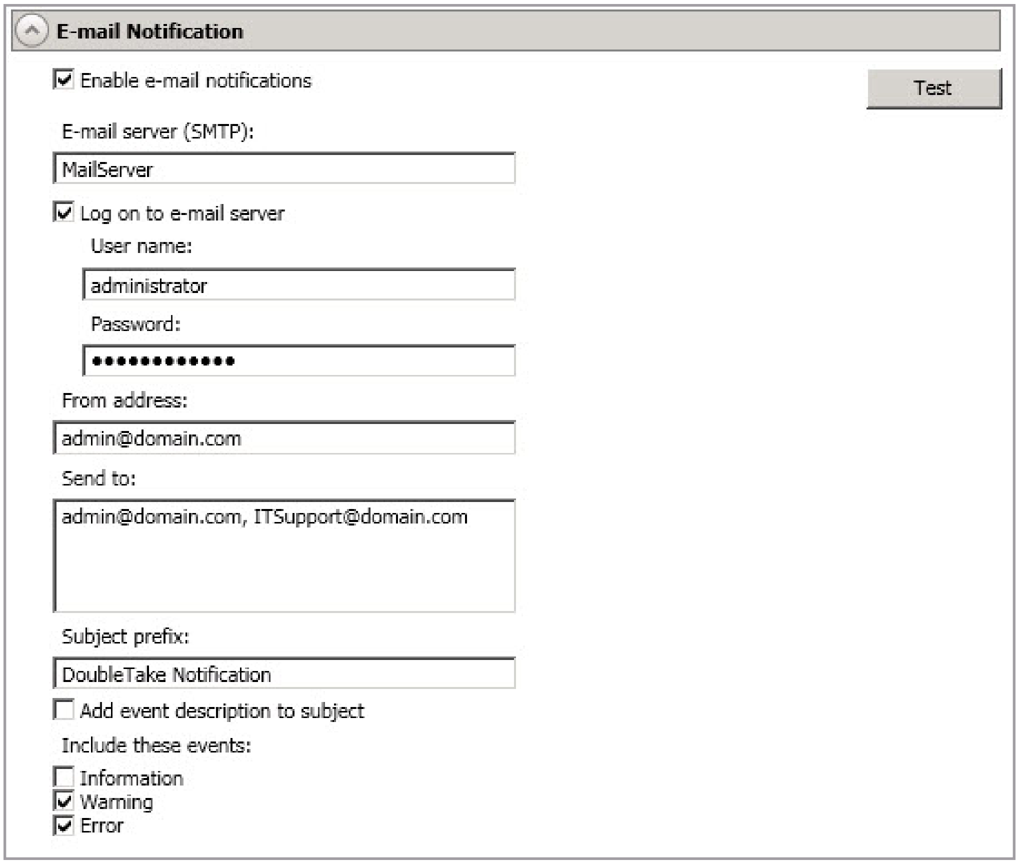
Write interval (minutes): 5

- **Logging folder**—Specify the directory where each of the log files in this section are stored. The default location is the directory where the Carbonite Availability program files are installed.
- **Messages & Alerts**—These settings apply to the service log file.
  - **Maximum size**—Specify the maximum size, in bytes, of the log file. The default size is 1048576 bytes (1 MB). Once the maximum has been reached, a new log file will be created.
  - **Maximum number of files**—Specify the maximum number of log files that are maintained. The default is 5, and the maximum is 999. Once the maximum has been reached, the oldest file will be overwritten.
- **Verification**—The verification log is created during the verification process and details which files were verified as well as the files that are synchronized.
  - **File name**—This field contains the base log file name for the verification process. The job type and a unique identifier will be prefixed to the base log file name. For example, since the default is DTVerify.log, the verification log for a files and folders job will be Files and Folders_123456abcdef DTVerify.log.
  - **Maximum size**—Specify the maximum size, in bytes, of the verification log file. The default is 1048576 bytes (1 MB).
  - **Append**—Enable the **Append** check box if you want to append each verification process to the same log file. If this check box is disabled, each verification process that is logged will overwrite the previous log file. By default, this option is enabled.
- **Statistics**—The statistics log maintains connection statistics such as mirror bytes in queue or replication bytes sent. This file is a binary file that is read by the DTStat utility. See the *Reference Guide* for details on DTStat.

- **File name**—This is the name of the statistics log file. The default file name is statistic.sts.
- **Maximum size**—Specify the maximum size, in bytes, of the statistics log file. The default is 10485760 bytes (10 MB). Once this maximum has been reached, the oldest data will be overwritten.
- **Write interval**—Specify how often, in minutes, Carbonite Availability writes to the statistics log file. The default is every 5 minutes.

E-mail notification configuration

You can email Carbonite Availability event messages to specific addresses using an SMTP mail server. The subject of the e-mail will contain an optional prefix, the server name where the message was logged, the message ID, and the severity level (information, warning, or error). The text of the event message will be displayed in the body of the e-mail message.



☑ Enable e-mail notifications

E-mail server (SMTP): MailServer

☑ Log on to e-mail server

User name: administrator

Password: ●●●●●●●●●●●●

From address: admin@domain.com

Send to: admin@domain.com, ITSupport@domain.com

Subject prefix: DoubleTake Notification

☐ Add event description to subject

Include these events:

☐ Information

☑ Warning

☑ Error

- **Enable e-mail notification**—This option enables the e-mail notification feature. Any specified notification settings will be retained if this option is disabled.
- **E-mail server**—Specify the name of your SMTP mail server.
- **Log on to e-mail server**—If your SMTP server requires authentication, enable this option and specify the **User name** and **Password** to be used for authentication. Your SMTP server must support the LOGIN authentication method to use this feature. If your server supports a different authentication method or does not support authentication, you may need to add the Carbonite Availability server as an authorized host for relaying e-mail messages. This option is not necessary if you are sending exclusively to e-mail addresses that the SMTP server is responsible for.
- **From address**—Specify the e-mail address that you want to appear in the From field of each Carbonite Availability e-mail message. The address is limited to 256 characters.
- **Send to**—Specify the e-mail addresses that each Carbonite Availability e-mail message should be sent to. Enter the addresses as a comma or semicolon separated list. Each address is limited to 256 characters. You can add up to 256 e-mail addresses.
- **Subject prefix** and **Add event description to subject**—The subject of each e-mail notification

will be in the format Subject Prefix : Server Name : Message Severity : Message ID : Message Description. The first and last components (Subject Prefix and Message Description) are optional. The subject line is limited to 100 characters.

If desired, enter unique text for the **Subject prefix** which will be inserted at the front of the subject line for each Carbonite Availability e-mail message. This will help distinguish Carbonite Availability messages from other messages. This field is optional.

If desired, enable **Add event description to subject** to have the description of the message appended to the end of the subject line. This field is optional.

When you modify your e-mail notification settings, you will receive a test e-mail summarizing your new settings. You can also test e-mail notification by clicking **Test**. By default, the test will be run from the machine where the console is running. If desired, you can send the test message to a different e-mail address by selecting **Send To** and entering a comma or semicolon separated list of addresses. Modify the **Message Text** up to 1024 characters, if necessary. Click **Send** to test the e-mail notification. The results will be displayed in a message box.

If an error occurs while sending an e-mail, a message will be generated. This message will not trigger another e-mail. Subsequent e-mail errors will not generate additional messages. When an e-mail is sent successfully, a message will then be generated. If another e-mail fails, one message will again be generated. This is a cyclical process where one message will be generated for each group of failed e-mail messages, one for each group of successful e-mail messages, one for the next group of failed messages, and so on.

If you start and then immediately stop the Double-Take service, you may not get e-mail notifications for the log entries that occur during startup.

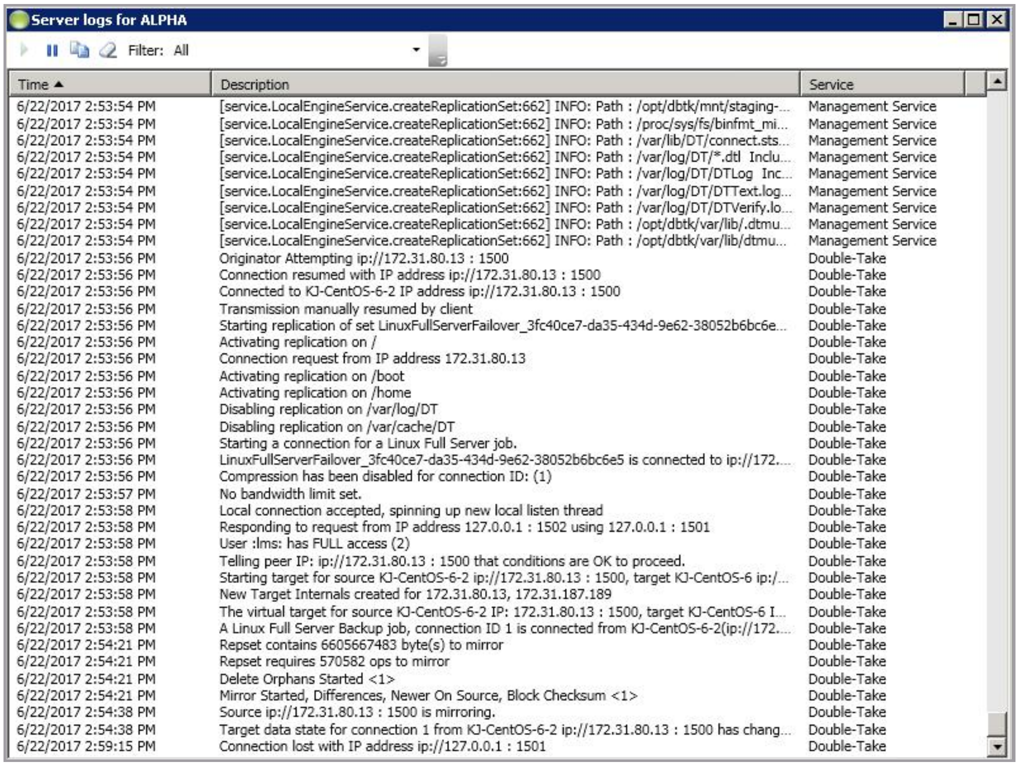
By default, most anti-virus software blocks unknown processes from sending traffic on port 25. You need to modify the blocking rule so that Carbonite Availability e-mail messages are not blocked.

Viewing server logs

You can view the engine and Management Service logs using either of these two methods.

- On the **Servers** page, highlight a server in the list and click **View Server Logs** from the toolbar.
- On the **Jobs** page, right-click a job and select **View Logs**. Select either the source server log or the target server log.

Separate logging windows allow you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple servers. When the Carbonite Replication Console is closed, all logging windows will automatically close.



```
Time ▲                   Description                                                                              Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /opt/dbtk/mnt/staging-...   Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /proc/sys/fs/binfmt_mi...   Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/lib/DT/connect.sts...  Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/*.dtl  Inclu... Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/DTLog  Inc...   Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/DTText.log...   Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /var/log/DT/DTVerify.lo...  Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /opt/dbtk/var/lib/.dtmu...  Management Service
6/22/2017 2:53:54 PM     [service.LocalEngineService.createReplicationSet:662] INFO: Path : /opt/dbtk/var/lib/dtmu...   Management Service
6/22/2017 2:53:56 PM     Originator Attempting ip://172.31.80.13 : 1500                                                 Double-Take
6/22/2017 2:53:56 PM     Connection resumed with IP address ip://172.31.80.13 : 1500                                    Double-Take
6/22/2017 2:53:56 PM     Connected to KJ-CentOS-6-2 IP address ip://172.31.80.13 : 1500                                 Double-Take
6/22/2017 2:53:56 PM     Transmission manually resumed by client                                                        Double-Take
6/22/2017 2:53:56 PM     Starting replication of set LinuxFullServerFailover_3fc40ce7-da35-434d-9e62-38052b6bc6e...     Double-Take
6/22/2017 2:53:56 PM     Activating replication on /                                                                    Double-Take
6/22/2017 2:53:56 PM     Connection request from IP address 172.31.80.13                                                Double-Take
6/22/2017 2:53:56 PM     Activating replication on /boot                                                                Double-Take
6/22/2017 2:53:56 PM     Activating replication on /home                                                                Double-Take
6/22/2017 2:53:56 PM     Disabling replication on /var/log/DT                                                           Double-Take
6/22/2017 2:53:56 PM     Disabling replication on /var/cache/DT                                                         Double-Take
6/22/2017 2:53:56 PM     Starting a connection for a Linux Full Server job.                                             Double-Take
6/22/2017 2:53:56 PM     LinuxFullServerFailover_3fc40ce7-da35-434d-9e62-38052b6bc6e5 is connected to ip://172....      Double-Take
6/22/2017 2:53:56 PM     Compression has been disabled for connection ID: (1)                                           Double-Take
6/22/2017 2:53:57 PM     No bandwidth limit set.                                                                        Double-Take
6/22/2017 2:53:58 PM     Local connection accepted, spinning up new local listen thread                                 Double-Take
6/22/2017 2:53:58 PM     Responding to request from IP address 127.0.0.1 : 1502 using 127.0.0.1 : 1501                  Double-Take
6/22/2017 2:53:58 PM     User :lms: has FULL access (2)                                                                 Double-Take
6/22/2017 2:53:58 PM     Telling peer IP: ip://172.31.80.13 : 1500 that conditions are OK to proceed.                   Double-Take
6/22/2017 2:53:58 PM     Starting target for source KJ-CentOS-6-2 ip://172.31.80.13 : 1500, target KJ-CentOS-6 ip:/...  Double-Take
6/22/2017 2:53:58 PM     New Target Internals created for 172.31.80.13, 172.31.187.189                                  Double-Take
6/22/2017 2:53:58 PM     The virtual target for source KJ-CentOS-6-2 IP: 172.31.80.13 : 1500, target KJ-CentOS-6 I...   Double-Take
6/22/2017 2:53:58 PM     A Linux Full Server Backup job, connection ID 1 is connected from KJ-CentOS-6-2(ip://172....   Double-Take
6/22/2017 2:54:21 PM     Repset contains 6605667483 byte(s) to mirror                                                   Double-Take
6/22/2017 2:54:21 PM     Repset requires 570582 ops to mirror                                                           Double-Take
6/22/2017 2:54:21 PM     Delete Orphans Started <1>                                                                     Double-Take
6/22/2017 2:54:21 PM     Mirror Started, Differences, Newer On Source, Block Checksum <1>                               Double-Take
6/22/2017 2:54:38 PM     Source ip://172.31.80.13 : 1500 is mirroring.                                                  Double-Take
6/22/2017 2:54:38 PM     Target data state for connection 1 from KJ-CentOS-6-2 ip://172.31.80.13 : 1500 has chang...    Double-Take
6/22/2017 2:59:15 PM     Connection lost with IP address ip://127.0.0.1 : 1501                                          Double-Take
```

The following table identifies the controls and the table columns in the **Server logs** window.

Start

This button starts the addition and scrolling of new messages in the window.

Pause

This button pauses the addition and scrolling of new messages in the window. This is only for the **Server logs** window. The messages are still logged to their respective files on the server.

Copy

This button copies the messages selected in the **Server logs** window to the Windows clipboard.

Clear

This button clears the **Server logs** window. The messages are not cleared from the respective files on the server. If you want to view all of the messages again, close and reopen the **Server logs** window.

Filter

From the drop-down list, you can select to view all log messages or only those messages from the Double-Take log or the Management Service log.

Time

This column in the table indicates the date and time when the message was logged.

Description

This column in the table displays the actual message that was logged.

Service

This column in the table indicates if the message is from the Double-Take log or the Management Service log.

Managing VMware servers

To manage your VMware servers, select **Go**, **Manage VMware Servers**. The **Manage VMware Server** page allows you to view, add, remove, or edit credentials for your VMware servers available in the console.

VMware Server

The name of the VMware server

Full Name

The full name of the VMware server

User Name

The user account being used to access the VMware server

Add VMware Server

Add a new VMware server. When prompted, specify the VMware server and a user account. If you are using a non-default port for your server, specify the server followed by a colon and then the port number, for example, 112.47.12.7:85. If your server name does not match the security certificate or the security certificate has expired, you will be prompted if you want to install the untrusted security certificate.

Remove Server

Remove the VMware server from the console.

Provide Credentials

Edit credentials for the selected VMware server. When prompted, specify a user account to access the VMware server.

Managing snapshots

Use the instructions below to manage the snapshots that Carbonite Availability has taken.

1. From the **Jobs** page, highlight the job and click **Manage Snapshots** in the toolbar.
2. You will see the list of snapshots, if any, associated with the job.
   - **Manual**—A user manually took this snapshot.
   - **Automatic**—Carbonite Availability automatically took this snapshot.
   - **Scheduled**—A periodic snapshot schedule triggered this snapshot.
   - **Deferred**—A periodic snapshot scheduled triggered this snapshot, although it did not occur at the specified interval because the job between the source and target was not in a good state.
   - **Test**—The test failover process took this snapshot.
   - **Coordinated**—A user took a coordinate snapshot.
   - **SQLClusterAutomatic**—The test failover process took this snapshot for a clustered SQL job.
3. Click **Take Snapshot** to create a new snapshot for the job.
4. If there is a snapshot that you no longer need, highlight it in the list and click **Delete**.
5. When you have completed your snapshot management, click **Close**.

Chapter 4 Files and folders protection

Create a files and folders job when you want to protect data. You can also use it to protect applications, such as Oracle or MySQL, however you will need to use your own customized failover and failback scripts to start and stop services during failover and failback. This job type does not protect a server's system state.

- *Files and folders requirements* on page 61—Files and folders protection includes specific requirements for this type of protection.
- *Creating a files and folders job* on page 66—This section includes step-by-step instructions for creating a files and folders job.
- *Managing and controlling files and folders jobs* on page 82—You can view status information about your files and folders jobs and learn how to control these jobs.
- *Failing over files and folders jobs* on page 99—Use this section when a failover condition has been met or if you want to failover manually.
- *Failback and restoration for files and folders jobs* on page 100—Use this section to determine if you want to failback and then restore or if you want to restore then failback.

Files and folders requirements

Use these requirements for files and folders protection.

- **Source and target servers**—The source and target servers can be a physical or virtual server running any of the following operating systems.
    - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
        - **Version**—5.9 through 5.11
        - **Kernel type for x86 (32-bit) architectures**—Default, SMP, Xen, PAE
        - **Kernel type for x86-64 (64-bit) architectures**—Default, SMP, Xen
        - **File system**—Ext3, Ext4, XFS
        - **Notes**—Oracle Enterprise Linux support is for the mainline kernel only, not the Unbreakable kernel.
    - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
        - **Version**—6.8 through 6.10
        - **Kernel type for x86 (32-bit) architectures**—Default
        - **Kernel type for x86-64 (64-bit) architectures**—Default
        - **File system**—Ext3, Ext4, XFS (64-bit only)
    - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
        - **Version**—7.3 through 7.5
        - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
        - **Kernel type for x86-64 (64-bit) architectures**—Default
        - **File system**—Ext3, Ext4, XFS
    - **Operating system**—SUSE Linux Enterprise
        - **Version**—11.2 through 11.4
        - **Kernel type for x86 (32-bit) architectures**—Default, Xen, XenPAE, VMI
        - **Kernel type for x86-64 (64-bit) architectures**—Default, Xen
        - **File system**—Ext3, XFS
    - **Operating system**—SUSE Linux Enterprise
        - **Version**—12.1 through 12.3
        - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
        - **Kernel type for x86-64 (64-bit) architectures**—Default
        - **File system**—Ext3, Ext4, XFS, Btrfs
        - **Notes**—If you are planning to convert an existing file system to Btrfs, you must delete any existing Carbonite Availability jobs and re-create them after converting to Btrfs.
    - **Operating system**—Ubuntu
        - **Version**—12.04.3, 12.04.4, and 12.04.5
        - **Kernel type for x86 (32-bit) architectures**—Generic

        - **Kernel type for x86-64 (64-bit) architectures**—Generic
        - **File system**—Ext2, Ext3, Ext4, XFS
    - **Operating system**—Ubuntu
        - **Version**—14.04.3, 14.04.4, and 14.04.5
        - **Kernel type for x86 (32-bit) architectures**—Generic
        - **Kernel type for x86-64 (64-bit) architectures**—Generic
        - **File system**—Ext2, Ext3, Ext4, XFS
    - **Operating system**—Ubuntu
        - **Version**—16.04.2, 16.04.3, and 16.04.4
        - **Kernel type for x86 (32-bit) architectures**—Generic
        - **Kernel type for x86-64 (64-bit) architectures**—Generic
        - **File system**—Ext2, Ext3, Ext4, XFS
    - **Operating system**—Ubuntu
        - **Version**—18.04.0
        - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
        - **Kernel type for x86-64 (64-bit) architectures**—Generic
        - **File system**—Ext2, Ext3, Ext4, XFS

> For all operating systems except Ubuntu, the kernel version must match the expected kernel for the specified release version. For example, if /etc/redhat-release declares the system to be a Redhat 7.5 system, the kernel that is installed must match that.
>
> Carbonite Availability does not support stacking filesystems, like eCryptFS.
>
> If Carbonite Availability does not have the driver binary files for the kernel you are using, they can be compiled automatically, but you need the build-essential package for them to be installed. Run apt-get install build-essential to install the build tools and then restart the DT service. This will build the driver from the source and load it.

- **Packages and services**—Each Linux server must have the following packages and services installed before you can install and use Carbonite Availability. See your operating system documentation for details on these packages and utilities.
    - sshd (or the package that installs sshd)
    - lsb
    - parted
    - dmidecode
    - scp
    - which
- **System memory**—The minimum system memory on each server is 1 GB.
- **Disk space for program files**—This is the amount of disk space needed for the Carbonite Availability program files. This is approximately 400 MB on each Linux server.

Make sure you have additional disk space for Carbonite Availability queuing, logging, and so on.

- **Server name**—Carbonite Availability includes Unicode file system support, but your server name must still be in ASCII format. Additionally, all Carbonite Availability servers must have a unique server name.
- **Protocols and networking**—Your servers must meet the following protocol and networking requirements.
    - Your servers must have TCP/IP with static IP addressing.
    - IPv4 is the only supported version.
    - If you are using Carbonite Availability over a WAN and do not have DNS name resolution, you will need to add the host names to the local hosts file on each server running Carbonite Availability.
    - Because of limitations in the way the Linux kernel handles IP address aliases, do not mix subnets on the eth0 network interface. Failover should not cause problems in this configuration, but you will lose IP addresses during failback. Therefore, if you must mix subnets on a single interface, use eth1 or higher.
    - Ubuntu Netplan is not a supported configuration. You must be running NetworkManager natively without Netplan. If you protect and failover a Linux server with Netplan, you will have to configure networking manually after failover.
- **NAT support**—Carbonite Availability supports NAT environments with the following caveats.
    - Only IPv4 is supported.
    - Only standalone servers are supported.
    - Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.
    - DNS failover and updates will depend on your configuration
        - Only the source or target can be behind a router, not both.
        - The DNS server must be routable from the target
- **Name resolution**—Your servers must have name resolution or DNS. The Carbonite Replication Console must be able to resolve the target, and the target must be able to resolve all source servers. For details on name resolution options, see your Linux documentation or online Linux resources.
- **Ports**—Port 1501 is used for localhost communication between the engine and management service and should be opened inbound and outbound for both TCP and UDP in iptables. Ports 1500, 1505, 1506, 6325, and 6326 are used for component communication and must be opened inbound and outbound for both TCP and UDP on any firewall that might be in use.
- **Security**—Carbonite Availability security is granted through membership in user groups. The groups can be local or LDAP (Lightweight Directory Access Protocol). A user must provide a valid local account that is a member of the Carbonite Availability security groups.

- **Docker**—Your source cannot be a Docker host.
- **VMware Tools**—Any VMWare guest running Carbonite Availability should have the appropriate VMWare Tools package installed.
- **Hard links**—If you have hard links outside of the data set you are protecting, and they link to files inside the data set you are protecting, Carbonite Availability will not mirror or replicate the hard links which could lead to differences on the target.
- **Supported configurations**—The following table identifies the supported configurations for a files and folders job.

Server Configuration	Description	Supported	Not Supported
One to one active/standby	You can protect a single source to a single target. The target has no production activity. The source is the only server actively replicating data.	X	
One to one active/active	You can protect a single source to a single target where each server acts as both a source and target actively replicating data to each other. Both servers are actively replicating data.	X	
Many to one	You can protect many source servers to one target server. Replication occurs from each source to the one target. This will consolidate your source servers to a single server.	X	
One to many	You can protect a single source to multiple target servers. The source is the only server actively replicating data. This will create redundant copies of your source.	X	
Chained	You can protect a single source to a single target, where the target then acts as a source, sending the same data from the original source to a final target server. The first source and the middle server are the only servers actively replicating data.	X	
Single server	You can protect a single source to itself allowing data to be replicated from one location to another on the same volume or to a separate volume on the same server.	X	
Standalone to standalone	Your servers can be in a standalone to standalone configuration.	X	
Standalone to cluster	Your servers cannot be in a standalone to cluster configuration.		X

Server Configuration	Description	Supported	Not Supported
Cluster to standalone	Your servers cannot be in a cluster to standalone configuration.		X
Cluster to cluster	Your servers cannot be in a cluster to cluster configuration.		X

Creating a files and folders job

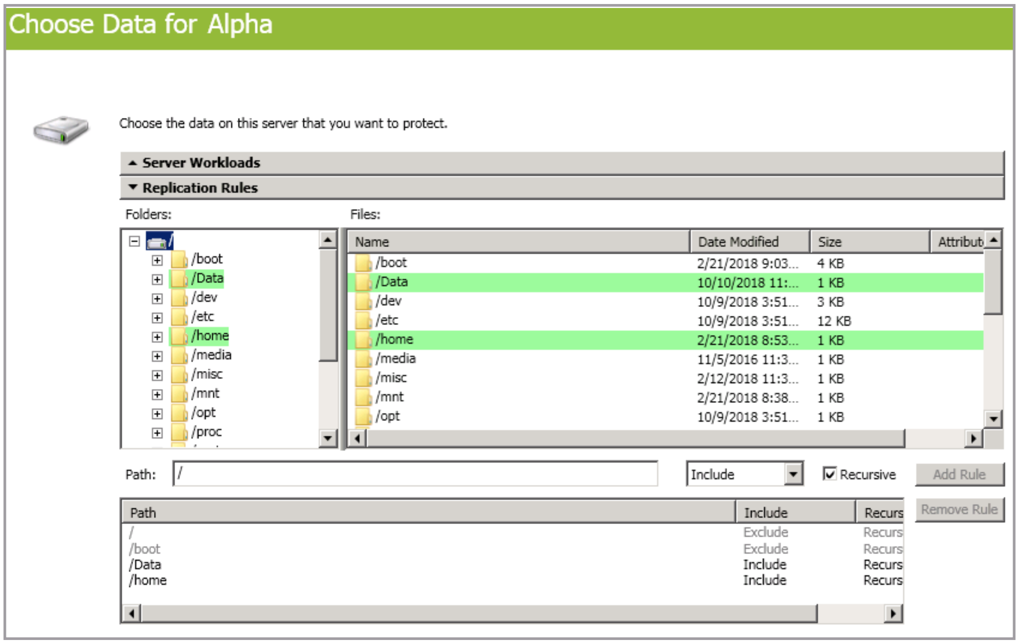
Use these instructions to create a files and folders job.

1. From the **Servers** page, right-click the server you want to protect and select **Protect**. You can also highlight a server, click **Create a New Job** in the toolbar, then select **Protect**.
2. Choose the type of workload that you want to protect. Under **Server Workloads**, in the **Workload types** pane, select **Files and Folders**. In the **Workload items** pane, you will see the volumes. Select the volumes you want to protect. You can select your files and folders in more detail in the **Replication Rules** section.

   Unsupported file systems will be displayed but will not be accessible.

   ---

3. To select your files and folders in more detail, click the **Replication Rules** heading and expand the volumes under **Folders**.



Choose Data for Alpha

Choose the data on this server that you want to protect.

Name	Date Modified	Size	Attribut
/boot	2/21/2018 9:03...	4 KB	
/Data	10/10/2018 11:...	1 KB	
/dev	10/9/2018 3:51...	3 KB	
/etc	10/9/2018 3:51...	12 KB	
/home	2/21/2018 8:53...	1 KB	
/media	11/5/2016 11:3...	1 KB	
/misc	2/12/2018 11:3...	1 KB	
/mnt	2/21/2018 8:38...	1 KB	
/opt	10/9/2018 3:51...	1 KB	

Path: / Include ☑ Recursive

Path	Include	Recurs
/	Exclude	Recurs
/boot	Exclude	Recurs
/Data	Include	Recurs
/home	Include	Recurs

Volumes and folders with a green highlight are included completely. Volumes and folders highlighted in light yellow are included partially, with individual files or folders included. If there is no highlight, no part of the volume or folder is included. To modify the items selected, highlight a volume, folder, or file and click **Add Rule**. Specify if you want to **Include** or **Exclude** the item. Also, specify if you want the rule to be recursive, which indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select **Recursive**, the rule will not be applied to subdirectories.

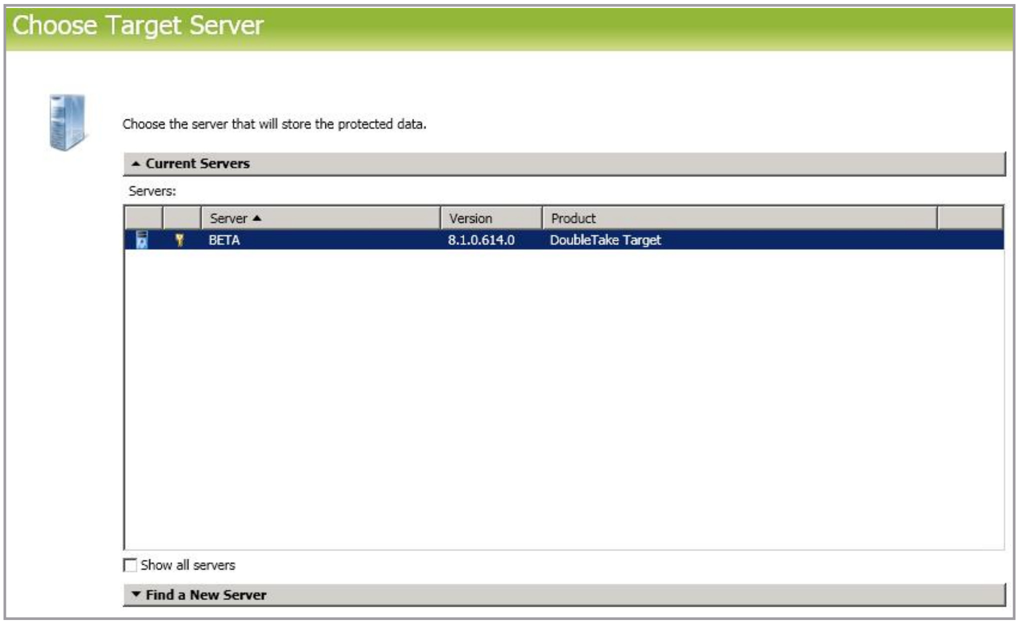
You can also enter wildcard rules, however you should do so carefully. Rules are applied to files that are closest in the directory tree to them. If you have rules that include multiple folders, an exclusion rule with a wild card will need to be added for each folder that it needs applied to. For example, if you want to exclude all .log files from /home and your rules include /home, /home/folder1, and /home/folder2, you would need to add the exclusion rule for the root and each subfolder rule. So you will need to add exclude rules for /home/*.log , /home/folder1/*.log, and /home/folder2/*.log.

If you need to remove a rule, highlight it in the list at the bottom and click **Remove Rule**. Be careful when removing rules. Carbonite Availability may create multiple rules when you are adding directories. For example, if you add /home/admin to be included in protection, then /home will be excluded. If you remove the /home exclusion rule, then the /home/admin rule will be removed also.

If you return to this page using the **Back** button in the job creation workflow, your **Workload Types** selection will be rebuilt, potentially overwriting any manual replication rules that you specified. If you do return to this page, confirm your **Workload Types** and **Replication Rules** are set to your desired settings before proceeding forward again.

---

4. Click **Next** to continue.
5. Choose your target server. This is the server that will store the replica data from the source.



Choose Target Server

Choose the server that will store the protected data.

▲ Current Servers

Servers:

Server ▲	Version	Product
BETA	8.1.0.614.0	DoubleTake Target

□ Show all servers

▼ Find a New Server

- **Current Servers**—This list contains the servers currently available in your console session. Servers that are not licensed for the workflow you have selected and those not applicable to the workload type you have selected will be filtered out of the list. Select your target server from the list. If the server you are looking for is not displayed, enable **Show all servers**. The servers in red are not available for the source server or workload type you have selected. Hover your mouse over an unavailable server to see a reason why this server is unavailable.
- **Find a New Server**—If the server you need is not in the **Current Servers** list, click the **Find a New Server** heading. From here, you can specify a server along with credentials for logging in to the server. If necessary, you can click **Browse** to select a server from a network drill-down list.

If you enter the target server's fully-qualified domain name, the Carbonite Replication Console will resolve the entry to the server short name. If that short name resides in two different domains, this could result in name resolution issues. In this case, enter the IP address of the server.

When specifying credentials for a new server, specify a user that is a member of the local dtadmin security group.

6. Click **Next** to continue.

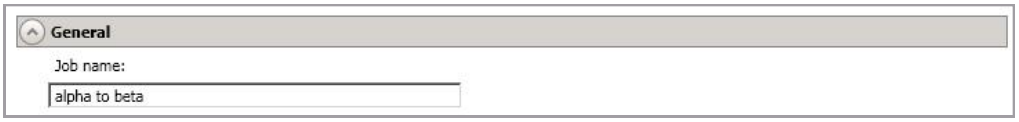
You may be prompted for a route from the target to the source. This route is used so the target can communicate with the source to build job options. This dialog box will be displayed only if needed.

7. You have many options available for your files and folders job. Configure those options that are applicable to your environment.

   Go to each page identified below to see the options available for that section of the **Set Options** page. After you have configured your options, continue with the next step on page 80.

   - *General* on page 68
   - *Failover Monitor* on page 69
   - *Failover Options* on page 71
   - *Failover Identity* on page 73
   - *Network Adapter Options* on page 74
   - *Mirror, Verify & Orphaned Files* on page 75
   - *Network Route* on page 77
   - *Path Mapping* on page 78
   - *Compression* on page 79
   - *Bandwidth* on page 80

General

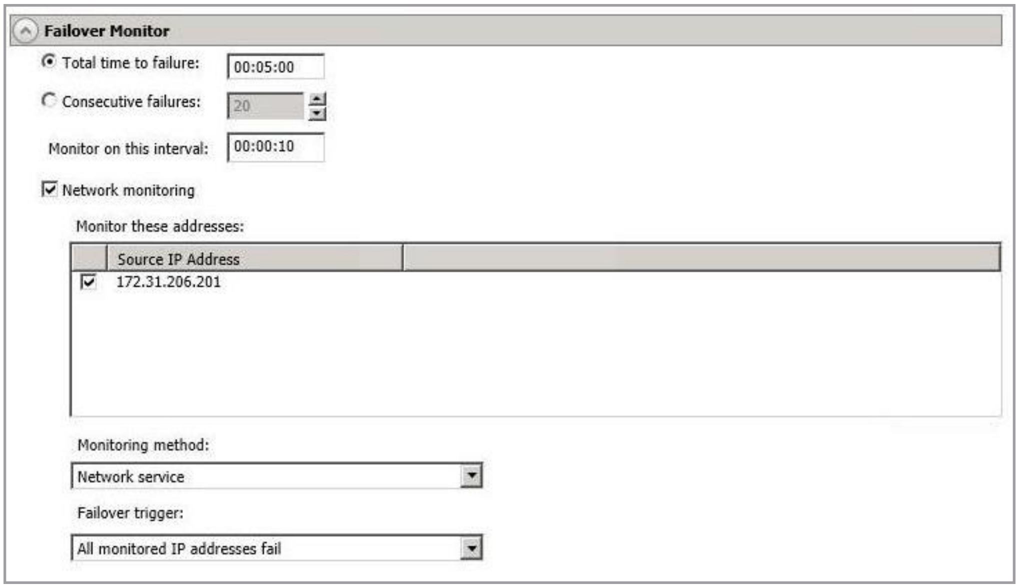


Job name:

alpha to beta

For the **Job name**, specify a unique name for your job.

Failover Monitor



Failover Monitor

● Total time to failure: 00:05:00

○ Consecutive failures: 20

Monitor on this interval: 00:00:10

☑ Network monitoring

Monitor these addresses:

	Source IP Address
☑	172.31.206.201

Monitoring method: Network service

Failover trigger: All monitored IP addresses fail

- **Total time to failure**—Specify, in hours:minutes:seconds, how long the target will keep trying to contact the source before the source is considered failed. This time is precise. If the total time has expired without a successful response from the source, this will be considered a failure.

  Consider a shorter amount of time for servers, such as a web server or order processing database, which must remain available and responsive at all times. Shorter times should be used where redundant interfaces and high-speed, reliable network links are available to prevent the false detection of failure. If the hardware does not support reliable communications, shorter times can lead to premature failover. Consider a longer amount of time for machines on slower networks or on a server that is not transaction critical. For example, failover would not be necessary in the case of a server restart.

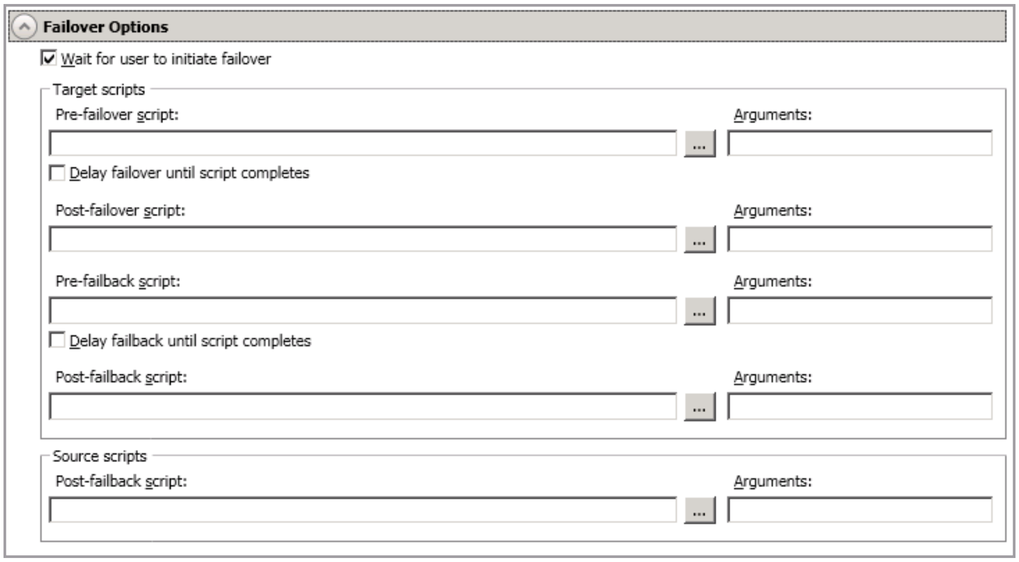
- **Consecutive failures—**Specify how many attempts the target will make to contact the source before the source is considered failed. For example, if you have this option set to 20, and your source fails to respond to the target 20 times in a row , this will be considered a failure.
- **Monitor on this interval**—Specify, in hours:minutes:seconds, how long to wait between attempts to contact the source to confirm it is online. This means that after a response (success or failure) is received from the source, Carbonite Availability will wait the specified interval time before contacting the source again. If you set the interval to 00:00:00, then a new check will be initiated immediately after the response is received.

  If you choose **Total time to failure**, do not specify a longer interval than failure time or your server will be considered failed during the interval period.

  If you choose **Consecutive failures**, your failure time is calculated by the length of time it takes your source to respond plus the interval time between each response, times the number of consecutive failures that can be allowed. That would be (response time + interval) * failure number. Keep in mind that timeouts from a failed check are included in the response time, so your failure time will not be precise.

- **Network monitoring**—With this option, the target will monitor the source using a network ping.
    - **Monitor these addresses**—Select each **Source IP Address** that you want the target to monitor. If you are using a NAT environment, do not select a private IP address on the source because the target cannot reach the source's private address, thus causing an immediate failure.
    - **Monitoring method**—This option determines the type of network ping used for failover monitoring.
        - **Network service**—Source availability will be tested by an ICMP ping to confirm the route is active.
        - **Replication service**—Source availability will be tested by a UDP ping to confirm the Double-Take service is active.
        - **Network and replication services**—Source availability will be tested by both an ICMP ping to confirm the route is active and a UDP ping to confirm the Double-Take service is active. If either monitoring method fails, failover will be triggered.
    - **Failover trigger**—If you are monitoring multiple IP addresses, specify when you want a failover condition to be triggered.
        - **One monitored IP address fails**—A failover condition will be triggered when any one of the monitored IP addresses fails. If each IP address is on a different subnet, you may want to trigger failover after one fails.
        - **All monitored IP addresses fail**—A failover condition will be triggered when all monitored IP addresses fail. If there are multiple, redundant paths to a server, losing one probably means an isolated network problem and you should wait for all IP addresses to fail.

Failover Options



Failover Options

☑ Wait for user to initiate failover

Target scripts

Pre-failover script: Arguments:

□ Delay failover until script completes

Post-failover script: Arguments:

Pre-failback script: Arguments:

□ Delay failback until script completes

Post-failback script: Arguments:

Source scripts

Post-failback script: Arguments:

- **Wait for user to initiate failover**—The failover process can wait for you to initiate it, allowing you to control when failover occurs. When a failure occurs, the job will wait in **Failover Condition Met** for you to manually initiate the failover process. Disable this option if you want failover to occur immediately when a failure occurs.
- **Scripts**—You can customize failover and failback by running scripts on the source and target. Scripts may contain any valid Linux command, executable, or shell script file. The scripts are processed using the same account running the Double-Take Management service. Examples of functions specified in scripts include stopping services on the target before failover because they may not be necessary while the target is standing in for the source, stopping services on the target that need to be restarted with the source's machine name and/or IP address, starting services or loading applications that are in an idle, standby mode waiting for failover to occur, notifying the administrator before and after failover or failback occurs, stopping services on the target after failback because they are no longer needed, stopping services on the target that need to be restarted with the target machine's original name and/or IP address, and so on. There are four types of failover and failback scripts that run on the target and one failback script that runs on the source.
  - **Pre-failover script**—This script runs on the target at the beginning of the failover process. Specify the full path and name of the script file.
  - **Post-failover script**—This script runs on the target at the end of the failover process. Specify the full path and name of the script file.
  - **Pre-failback script** —This script runs on the target at the beginning of the failback process. Specify the full path and name of the script file.
  - **Post-failback script**—This script runs on the target or source at the end of the failback process. Specify the full path and name of the script file.
  - **Arguments**—Specify a comma-separated list of valid arguments required to execute the script.
  - **Delay until script completes**—Enable this option if you want to delay the failover or failback process until the associated script has completed. If you select this option,

make sure your script handles errors, otherwise the failover or failback process may never complete if the process is waiting on a script that cannot complete.

NFS exports and Samba shares must be configured for failover through the failover scripts or created manually on the target after failover. If you want to script it, use the following steps.

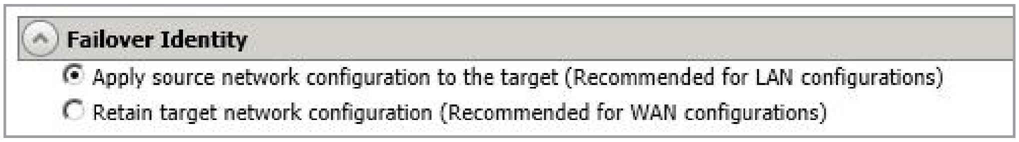
1. Stop and restart the NFS and/or Samba service on the source. The Carbonite Availability service must be running before the NFS or Samba service in order for replication operations to be captured.
2. On your target, set the NFS and/or Samba and WinBind services to manual startup. This allows the failover script to control when the service starts on the target.
3. Make sure the data you are protecting includes /etc/exports for NFS and /etc/SAMBA/samba_conf for Samba and the shared data.
4. Add the following to your post-failover script. If you are only using NFS, add only the first line. If you are only using Samba, add only the second line. If you are using both, add both lines.

   service nfs start

   service smb start

After failover, the services you added to your script will automatically be started by the script. If your clients see a stale file handle error message when attempting to access an export, they will need to reconnect to it.

Failover Identity



Failover Identity

● Apply source network configuration to the target (Recommended for LAN configurations)

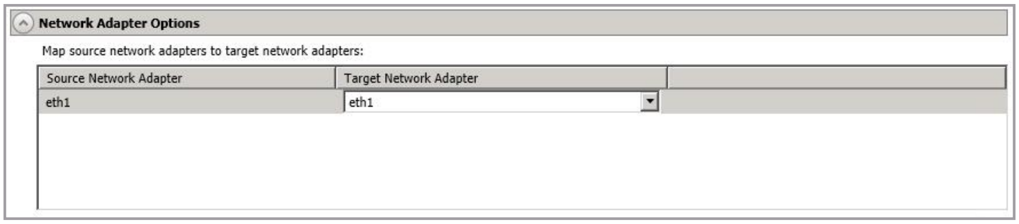
○ Retain target network configuration (Recommended for WAN configurations)

- **Apply source network configuration to the target**—If you select this option, your source IP addresses will failover to the target. If your target is on the same subnet as the source (typical of a LAN environment), you should select this option. Do not select this option if you are using a NAT environment that has a different subnet on the other side of the router.

  Do not apply the source network configuration to the target in a WAN environment unless you have a VPN infrastructure so that the source and target can be on the same subnet, in which case IP address failover will work the same as a LAN configuration. If you do not have a VPN, you will have to reconfigure the routers by moving the source's subnet from the source's physical network to the target's physical network. There are a number of issues to consider when designing a solution that requires router configuration to achieve IP address failover. Since the route to the source's subnet will be changed at failover, the source server must be the only system on that subnet, which in turn requires all server communications to pass through a router. Additionally, it may take several minutes or even hours for routing tables on other routers throughout the network to converge.

- **Retain target network configuration**—If you select this option, the target will retain all of its original IP addresses. If your target is on a different subnet (typical of a WAN or NAT environment), you should select this option.

Network Adapter Options



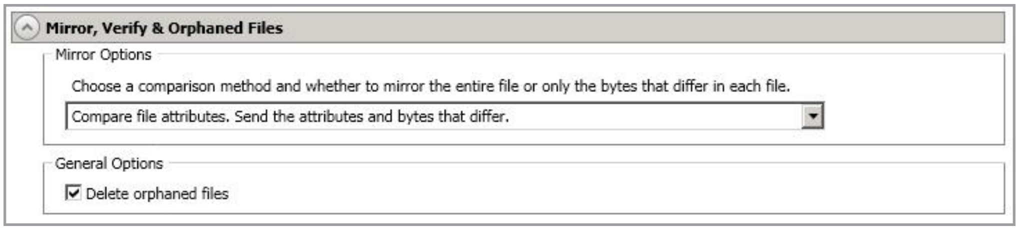
Network Adapter Options

Map source network adapters to target network adapters:

Source Network Adapter	Target Network Adapter	
eth1	eth1	

For **Map source network adapters to target network adapters**, specify how you want the IP addresses associated with each NIC on the source to be mapped to a NIC on the target. Do not mix public and private networks.

Mirror, Verify & Orphaned Files



Mirror, Verify & Orphaned Files

Mirror Options

Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.

Compare file attributes. Send the attributes and bytes that differ.

General Options

☑ Delete orphaned files

- **Mirror Options**—Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.
  - **Do not compare files. Send the entire file.**—Carbonite Availability will not perform any comparisons between the files on the source and target. All files will be mirrored to the target, sending the entire file. This option requires no time for comparison, but it can be slower because it sends the entire file. However, it is useful for configurations that have large data sets with millions of small files that are frequently changing and it is more efficient to send the entire file. You may also need to use this option if configuration management policies require sending the entire file.
  - **Compare file attributes. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and will mirror only the attributes and bytes that are different. This option is the fastest comparison method and fastest mirror option. Files that have not changed can be easily skipped. Also files that are open and require a checksum mirror can be compared.
  - **Compare file attributes and data. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and the file data and will mirror only the attributes and bytes that are different. This comparison method is not as fast because every file is compared, regardless of whether the file has changed or is open. However, sending only the attributes and bytes that differ is the fasted mirror option.

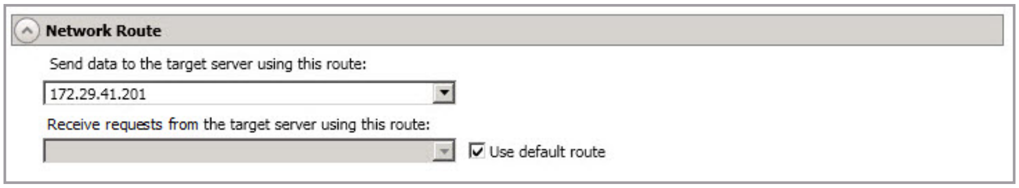
  If a file is small enough that mirroring the entire file is faster than comparing it and then mirroring it, Carbonite Availability will automatically mirror the entire file.

- **General Options**—Choose your general mirroring options.
  - **Delete orphaned files**—An orphaned file is a file that exists in the replica data on the target, but does not exist in the protected data on the source. This option specifies if orphaned files should be deleted on the target.

    Orphaned file configuration is a per target configuration. All jobs to the same target will have the same orphaned file configuration.

    If delete orphaned files is enabled, carefully review any replication rules that use wildcard definitions. If you have specified wildcards to be excluded from protection, files matching those wildcards will also be excluded from orphaned file processing and will not be deleted from the target. However, if you have specified wildcards to be included in your protection, those files that fall outside the wildcard inclusion rule will be considered orphaned files and will be deleted from the target.

Network Route



Network Route

Send data to the target server using this route:

172.29.41.201

Receive requests from the target server using this route:

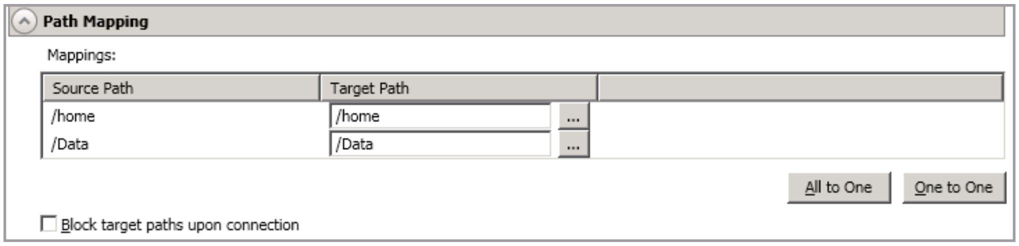
☑ Use default route

- **Send data to the target server using this route**—By default, Carbonite Availability will select a target route for transmissions. If desired, specify an alternate route on the target that the data will be transmitted through. This allows you to select a different route for Carbonite Availability traffic. For example, you can separate regular network traffic and Carbonite Availability traffic on a machine with multiple IP addresses. You can also select or manually enter a public IP address (which is the public IP address of the server's router) if you are using a NAT environment.

  If you change the IP address on the target which is used for the target route, you will be unable to edit the job. If you need to make any modifications to the job, it will have to be deleted and re-created.

- **Receive requests from the target server using this route**—By default, Carbonite Availability will select a route from the target for command and status requests. If desired, specify an alternate route on the target that the commands will be transmitted from. This allows you to select a different route for Carbonite Availability management communication. You can also manually enter a public IP address (which is the public IP address of the server's router) if you are using a NAT environment.
- **Use default route**—Select this option to disable the drop-down list that allows you to select the route from the target server. When this option is enabled, the default route will automatically be used.

Path Mapping


Path Mapping

Mappings:

Source Path	Target Path
/home	/home
/Data	/Data

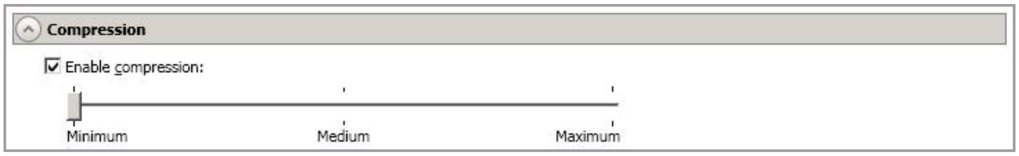
□ Block target paths upon connection

- **Mappings**—Specify the location on the target where the replica of the source data will be stored. By default, the replica source data will be stored in the same directory structure on the target, in a one to one configuration. Make sure you update this location if you are protecting multiple sources or jobs to the same target. You have two pre-defined locations as well as a custom option that allows you to set your path.
  - **All To One**—Click this button to set the mapping so that the replica source data will be stored on a single volume on the target. The pre-defined path is /source_ name/volume_name. If you are protecting multiple volumes on the source, each volume would be stored on the same volume on the target.
  - **One To One**—Click this button to set the mapping so that the replica source data will be stored in the same directory structure on the target. For example, /data and /home will be stored in /data and /home, respectively, on the target.
  - **Custom Location**—If the pre-defined options do not store the data in a location that is appropriate for your network operations, you can specify your own custom location where the replica source data will be stored. Click the **Target Path** and edit it, selecting the appropriate location.

  If you are protecting system state data , you must select the **All to One** mapping or specify a customized location in order to avoid sharing violations. Keep in mind that this mapping will avoid sharing violations on the target, however during a restoration, you will get sharing violations on the source because the restoration mapping is one to one and your system state files will be in use on the source you are restoring to. In this case, restoration will never complete. If you will need to restore data and you must protect system state data, you should use a full server job.

- **Block target paths upon connection**—You can block writing to the replica source data located on the target. This keeps the data from being changed outside of Carbonite Availability processing. Any target paths that are blocked will be unblocked automatically during the failover process so that users can modify data after failover. During restoration, the paths are automatically blocked again. If you failover and failback without performing a restoration, the target paths will remain unblocked.

Compression


Compression

☑ Enable compression:

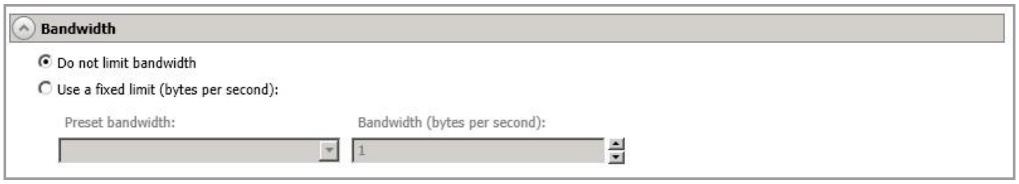
To help reduce the amount of bandwidth needed to transmit Carbonite Availability data, compression allows you to compress data prior to transmitting it across the network. In a WAN environment this provides optimal use of your network resources. If compression is enabled, the data is compressed before it is transmitted from the source. When the target receives the compressed data, it decompresses it and then writes it to disk. You can set the level from **Minimum** to **Maximum** to suit your needs.

Keep in mind that the process of compressing data impacts processor usage on the source. If you notice an impact on performance while compression is enabled in your environment, either adjust to a lower level of compression, or leave compression disabled. Use the following guidelines to determine whether you should enable compression.

- If data is being queued on the source at any time, consider enabling compression.
- If the server CPU utilization is averaging over 85%, be cautious about enabling compression.
- The higher the level of compression, the higher the CPU utilization will be.
- Do not enable compression if most of the data is inherently compressed. Many image (.jpg, .gif) and media (.wmv, .mp3, .mpg) files, for example, are already compressed. Some images files, such as .bmp and .tif, are decompressed, so enabling compression would be beneficial for those types.
- Compression may improve performance even in high-bandwidth environments.
- Do not enable compression in conjunction with a WAN Accelerator. Use one or the other to compress Carbonite Availability data.

All jobs from a single source connected to the same IP address on a target will share the same compression configuration.

Bandwidth



Bandwidth

◉ Do not limit bandwidth

○ Use a fixed limit (bytes per second):

Preset bandwidth:

Bandwidth (bytes per second): 1

Bandwidth limitations are available to restrict the amount of network bandwidth used for Carbonite Availability data transmissions. When a bandwidth limit is specified, Carbonite Availability never exceeds that allotted amount. The bandwidth not in use by Carbonite Availability is available for all other network traffic.

All jobs from a single source connected to the same IP address on a target will share the same bandwidth configuration.

- **Do not limit bandwidth**—Carbonite Availability will transmit data using 100% bandwidth availability.
- **Use a fixed limit**—Carbonite Availability will transmit data using a limited, fixed bandwidth. Select a **Preset bandwidth** limit rate from the common bandwidth limit values. The **Bandwidth** field will automatically update to the bytes per second value for your selected bandwidth. This is the maximum amount of data that will be transmitted per second. If desired, modify the bandwidth using a bytes per second value. The minimum limit should be 3500 bytes per second.

8. Click **Next** to continue.
9. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

   Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

   If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

   After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Migrate Reference Guide* for details on the various Carbonite Availability log files.
10. Once your servers have passed validation and you are ready to establish protection, click **Finish**,

and you will automatically be taken to the **Jobs** page.

Jobs in a NAT environment may take longer to start.

Once a job is created, do not change the name of underlying hardware components used in the job. For example, volume and datastore names or network adapter and virtual switch names. Any component used by name in your job must continue to use that name throughout the lifetime of job. If you must change a name, you will need to delete the job and re-create it using the new component name.

Managing and controlling files and folders jobs

Click **Jobs** from the main Carbonite Replication Console toolbar. The **Jobs** page allows you to view status information about your jobs. You can also control your jobs from this page.

The jobs displayed in the right pane depend on the server group folder selected in the left pane. Every job for each server in your console session is displayed when the **Jobs on All Servers** group is selected. If you have created and populated server groups (see *Managing servers* on page 24), then only the jobs associated with the server or target servers in that server group will be displayed in the right pane.

- *Overview job information displayed in the top right pane* on page 82
- *Detailed job information displayed in the bottom right pane* on page 85
- *Job controls* on page 87

Overview job information displayed in the top right pane

The top pane displays high-level overview information about your jobs. You can sort the data within a column in ascending and descending order. You can also move the columns to the left or right of each other to create your desired column order. The list below shows the columns in their default left to right order.

If you are using server groups, you can filter the jobs displayed in the top right pane by expanding the **Server Groups** heading and selecting a server group.

Column 1 (Blank)

The first blank column indicates the state of the job.

✓ A green circle with a white checkmark indicates the job is in a healthy state. No action is required.

⚠ A yellow triangle with a black exclamation point indicates the job is in a pending or warning state. This icon is also displayed on any server groups that you have created that contain a job in a pending or warning state. Carbonite Availability is working or waiting on a pending process or attempting to resolve the warning state.

✗ A red circle with a white X indicates the job is in an error state. This icon is also displayed on any server groups that you have created that contain a job in an error state. You will need to investigate and resolve the error.

? The job is in an unknown state.

Job

The name of the job

Source Server

The name of the source. This could be the name or IP address of your source.

Target Server

The name of the target. This could be the name or IP address of your target.

Job Type

Each job type has a unique job type name. This job is a Files and Folders job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the job details. Keep in mind that **Idle** indicates console to server activity is idle, not that your servers are idle.

Mirror Status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.
- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Restoring**—Data is being restored from the target to the source.
- **Unknown**—The console cannot determine the status.

Replication Status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Transmit Mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Operating System	The job type operating system

Detailed job information displayed in the bottom right pane

The details displayed in the bottom pane provide additional information for the job highlighted in the top pane. You can expand or collapse the bottom pane by clicking on the **Job Highlights** heading.

Name

The name of the job

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Restore Required**—The data on the source and target do not match because of a failover condition. Restore the data from the target back to the source. If you want to discard the changes on the target, you can remirror to resynchronize the source and target.
- **Snapshot Reverted**—The data on the source and target do not match because a snapshot has been applied on the target. Restore the data from the target back to the source. If you want to discard the changes on the target, you can remirror to resynchronize the source and target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the recovery point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the recovery point latency time.

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent (compressed)

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Connected since

The date and time indicating when the current job was started.

Recent activity

Displays the most recent activity for the selected job, along with an icon indicating the success or failure of the last initiated activity. Click the link to see a list of recent activities for the selected job. You can highlight an activity in the list to display additional details about the activity.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Job controls

You can control your job through the toolbar buttons available on the **Jobs** page. If you select multiple jobs, some of the controls will apply only to the first selected job, while others will apply to all of the selected jobs. For example, **View Job Details** will only show details for the first selected job, while **Stop** will stop protection for all of the selected jobs.

If you want to control just one job, you can also right click on that job and access the controls from the pop-up menu.

View Job Details

This button leaves the **Jobs** page and opens the **View Job Details** page.

Edit Job Properties

This button leaves the **Jobs** page and opens the **EditJob Properties** page.

Delete

Stops (if running) and deletes the selected jobs.

Provide Credentials

Changes the login credentials that the job (which is on the target machine) uses to authenticate to the servers in the job. This button opens the Provide Credentials dialog box where you can specify the new account information and which servers you want to update.

View Recent Activity

Displays the recent activity list for the selected job. Highlight an activity in the list to display additional details about the activity.

Start

Starts or resumes the selected jobs.

If you have previously stopped protection, the job will restart mirroring and replication.

If you have previously paused protection, the job will continue mirroring and replication from where it left off, as long as the Carbonite Availability queue was not exhausted during the time the job was paused. If the Carbonite Availability queue was exhausted during the time the job was paused, the job will restart mirroring and replication.

Also if you have previously paused protection, all jobs from the same source to the same IP address on the target will be resumed.

Pause

Pauses the selected jobs. Data will be queued on the source while the job is paused. Failover monitoring will continue while the job is paused.

All jobs from the same source to the same IP address on the target will be paused.

Stop

Stops the selected jobs. The jobs remain available in the console, but there will be no mirroring or replication data transmitted from the source to the target. Mirroring and replication data will not be queued on the source while the job is stopped, requiring a remirror when the job is restarted. The type of remirror will depend on your job settings. Failover monitoring will continue while the job is stopped.

Take Snapshot

Snapshots are not applicable to files and folders jobs.

Manage Snapshots

Snapshots are not applicable to files and folders jobs.

Failover or Cutover

Starts the failover process. See *Failing over files and folders jobs* on page 99 for the process and details of failing over a files and folders job.

Failback

Starts the failback process. See *Failback and restoration for files and folders jobs* on page 100 for the process and details of failing back a files and folders job.

Restore

Starts the restoration process. See *Failback and restoration for files and folders jobs* on page 100 for the process and details of restoring a files and folders job.

Reverse

Reverses protection. Reverse protection does not apply to files and folders jobs.

Undo Failover or Cutover

Cancels a test failover by undoing it. Undo failover does not apply to files and folders jobs.

View Job Log

Opens the job log. On the right-click menu, this option is called **View Logs**, and you have the option of opening the job log, source server log, or target server log.

Other Job Actions

Opens a small menu of other job actions. These job actions are not available for Linux jobs.

Generate Activity Report

For all Windows jobs except files and folders, you can create a basic failover report. This is the same report created by Get-DtLatestFailoverReport and Get-DtAllFailoverReports from the Carbonite Availability PowerShell module. The report will be located on the target in the \Service\Reports directory where Carbonite Availability is installed. For Linux jobs, you must use PowerShell.

Filter

Select a filter option from the drop-down list to only display certain jobs. You can display **Healthy jobs**, **Jobs with warnings**, or **Jobs with errors**. To clear the filter, select **All jobs**. If you have created and populated server groups, then the filter will only apply to the jobs associated with the server or target servers in that server group. See *Managing servers* on page 24.

Search

Allows you to search the source or target server name for items in the list that match the criteria you have entered.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Viewing files and folders job details

From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.

Review the following table to understand the detailed information about your job displayed on the **View Job Details** page.

Job name

The name of the job

Job type

Each job type has a unique job type name. This job is a Files and Folders job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Health

✓ The job is in a healthy state.

⚠ The job is in a warning state.

✖ The job is in an error state.

? The job is in an unknown state.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the job details.

Connection ID

The incremental counter used to number connections. The number is incremented when a connection is created. The counter is reset if there are no existing jobs and the Double-Take service is restarted.

Transmit mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Restore Required**—The data on the source and target do not match because of a failover condition. Restore the data from the target back to the source. If you want to discard the changes on the target, you can remirror to resynchronize the source and target.
- **Snapshot Reverted**—The data on the source and target do not match because a snapshot has been applied on the target. Restore the data from the target back to the source. If you want to discard the changes on the target, you can remirror to resynchronize the source and target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Target route

The IP address on the target used for Carbonite Availability transmissions.

Compression

- **On** / **Level**—Data is compressed at the level specified.
- **Off—**Data is not compressed.

Encryption

- **On—**Data is being encrypted before it is sent from the source to the target.
- **Off—**Data is not being encrypted before it is sent from the source to the target.

Bandwidth limit

If bandwidth limiting has been set, this statistic identifies the limit. The keyword **Unlimited** means there is no bandwidth limit set for the job.

Connected since

The source server date and time indicating when the current job was started. This field is blank, indicating that a TCP/IP socket is not present, when the job is waiting on transmit options or if the transmission has been stopped. This field will maintain the date and time, indicating that a TCP/IP socket is present, when transmission has been paused.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Mirror status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.
- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Restoring**—Data is being restored from the target to the source.
- **Unknown**—The console cannot determine the status.

Mirror percent complete

The percentage of the mirror that has been completed

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent compressed

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the recovery point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the recovery point latency time.

Mirror start time

The UTC time when mirroring started

Mirror end time

The UTC time when mirroring ended

Total time for last mirror

The length of time it took to complete the last mirror process

Validating a files and folders job

Over time, you may want to confirm that any changes in your network or environment have not impacted your Carbonite Availability job. Use these instructions to validate an existing job.

1. From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.
2. In the **Tasks** area on the right on the **View Job Details** page, click **Validate job properties**.
3. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

   Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

   Validation checks for an existing job are logged to the job log on the target server.
4. Once your servers have passed validation, click **Close**.

Editing a files and folders job

Use these instructions to edit a files and folders job.

1. From the **Jobs** page, highlight the job and click **Edit Job Properties** in the toolbar. (You will not be able to edit a job if you have removed the source of that job from your Carbonite Replication Console session or if you only have Carbonite Availability monitor security access.)
2. You will see the same options available for your files and folders job as when you created the job, but you will not be able to edit all of them. If desired, edit those options that are configurable for an existing job. See *Creating a files and folders job* on page 66 for details on each job option.

   Changing some options may require Carbonite Availability to automatically disconnect, reconnect, and remirror the job.

   If you have specified replication rules that exclude a volume at the root, that volume will be incorrectly added as an inclusion if you edit the job after it has been established. If you need to edit your job, modify the replication rules to make sure they include the proper inclusion and exclusion rules that you want.

3. If you want to modify the workload items or replication rules for the job, click **Edit workload or replication rules**. Modify the **Workload item** you are protecting, if desired. Additionally, you can modify the specific **Replication Rules** for your job.

   Click **OK** to return to the **Edit Job Properties** page.

   If you remove data from your workload and that data has already been sent to the target, you will need to manually remove that data from the target. Because the data you removed is no longer included in the replication rules, Carbonite Availability orphan file detection cannot remove the data for you. Therefore, you have to remove it manually.

4. Click **Next** to continue.
5. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

   Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

   If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

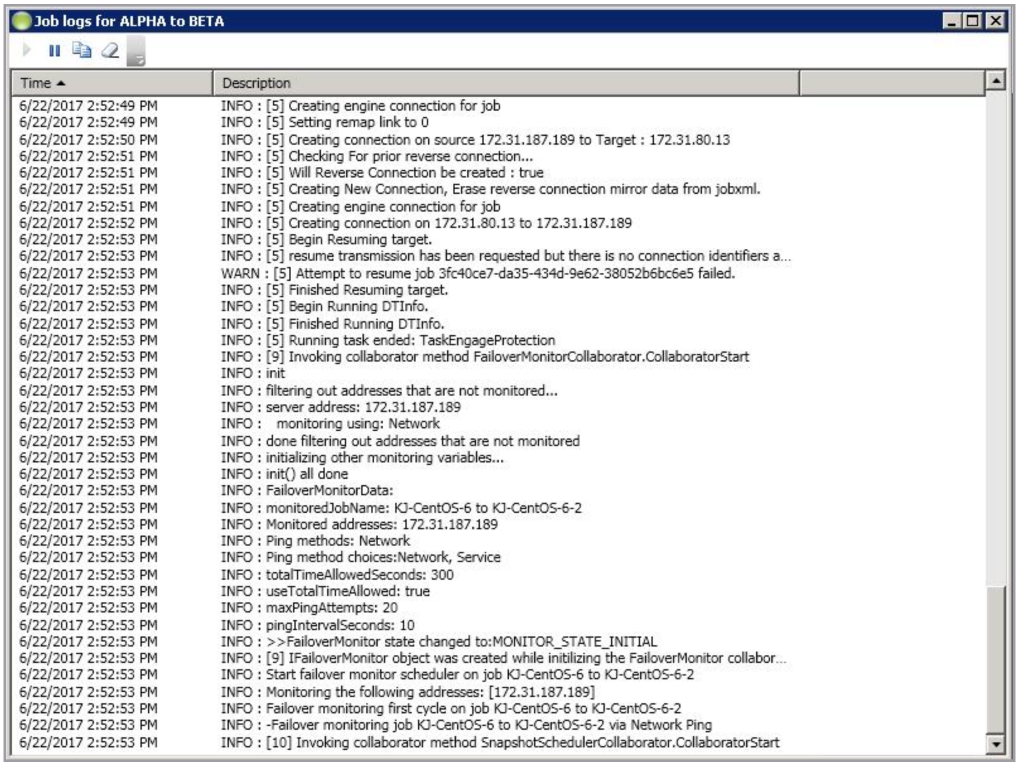
After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Migrate Reference Guide* for details on the various Carbonite Availability log files.

6. Once your servers have passed validation and you are ready to update your job, click **Finish**.

Viewing a files and folders job log

You can view a job log file through the Carbonite Replication Console by selecting **View Job Log** from the toolbar on the **Jobs** page. Separate logging windows allow you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple jobs. When the Carbonite Replication Console is closed, all logging windows will automatically close.

Because the job log window communicates with the target server, if the console loses communication with the target server after the job log window has already been opened, the job log window will display an error.



```
Time ▲                  Description
6/22/2017 2:52:49 PM    INFO : [5] Creating engine connection for job
6/22/2017 2:52:49 PM    INFO : [5] Setting remap link to 0
6/22/2017 2:52:50 PM    INFO : [5] Creating connection on source 172.31.187.189 to Target : 172.31.80.13
6/22/2017 2:52:51 PM    INFO : [5] Checking For prior reverse connection...
6/22/2017 2:52:51 PM    INFO : [5] Will Reverse Connection be created : true
6/22/2017 2:52:51 PM    INFO : [5] Creating New Connection, Erase reverse connection mirror data from jobxml.
6/22/2017 2:52:51 PM    INFO : [5] Creating engine connection for job
6/22/2017 2:52:52 PM    INFO : [5] Creating connection on 172.31.80.13 to 172.31.187.189
6/22/2017 2:52:53 PM    INFO : [5] Begin Resuming target.
6/22/2017 2:52:53 PM    INFO : [5] resume transmission has been requested but there is no connection identifiers a...
6/22/2017 2:52:53 PM    WARN : [5] Attempt to resume job 3fc40ce7-da35-434d-9e62-38052b6bc6e5 failed.
6/22/2017 2:52:53 PM    INFO : [5] Finished Resuming target.
6/22/2017 2:52:53 PM    INFO : [5] Begin Running DTInfo.
6/22/2017 2:52:53 PM    INFO : [5] Finished Running DTInfo.
6/22/2017 2:52:53 PM    INFO : [5] Running task ended: TaskEngageProtection
6/22/2017 2:52:53 PM    INFO : [9] Invoking collaborator method FailoverMonitorCollaborator.CollaboratorStart
6/22/2017 2:52:53 PM    INFO : init
6/22/2017 2:52:53 PM    INFO : filtering out addresses that are not monitored...
6/22/2017 2:52:53 PM    INFO : server address: 172.31.187.189
6/22/2017 2:52:53 PM    INFO :   monitoring using: Network
6/22/2017 2:52:53 PM    INFO : done filtering out addresses that are not monitored
6/22/2017 2:52:53 PM    INFO : initializing other monitoring variables...
6/22/2017 2:52:53 PM    INFO : init() all done
6/22/2017 2:52:53 PM    INFO : FailoverMonitorData:
6/22/2017 2:52:53 PM    INFO : monitoredJobName: KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM    INFO : Monitored addresses: 172.31.187.189
6/22/2017 2:52:53 PM    INFO : Ping methods: Network
6/22/2017 2:52:53 PM    INFO : Ping method choices:Network, Service
6/22/2017 2:52:53 PM    INFO : totalTimeAllowedSeconds: 300
6/22/2017 2:52:53 PM    INFO : useTotalTimeAllowed: true
6/22/2017 2:52:53 PM    INFO : maxPingAttempts: 20
6/22/2017 2:52:53 PM    INFO : pingIntervalSeconds: 10
6/22/2017 2:52:53 PM    INFO : >>FailoverMonitor state changed to:MONITOR_STATE_INITIAL
6/22/2017 2:52:53 PM    INFO : [9] IFailoverMonitor object was created while initilizing the FailoverMonitor collabor...
6/22/2017 2:52:53 PM    INFO : Start failover monitor scheduler on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM    INFO : Monitoring the following addresses: [172.31.187.189]
6/22/2017 2:52:53 PM    INFO : Failover monitoring first cycle on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM    INFO : -Failover monitoring job KJ-CentOS-6 to KJ-CentOS-6-2 via Network Ping
6/22/2017 2:52:53 PM    INFO : [10] Invoking collaborator method SnapshotSchedulerCollaborator.CollaboratorStart
```

The following table identifies the controls and the table columns in the **Job logs** window.

Start ▶

This button starts the addition and scrolling of new messages in the window.

Pause ❚❚

This button pauses the addition and scrolling of new messages in the window. This is only for the **Job logs** window. The messages are still logged to their respective files on the server.

Copy

This button copies the messages selected in the **Job logs** window to the Windows clipboard.

Clear

This button clears the **Job logs** window. The messages are not cleared from the respective files on the server. If you want to view all of the messages again, close and reopen the **Job logs** window.

Time

This column in the table indicates the date and time when the message was logged.

Description

This column in the table displays the actual message that was logged.

Failing over files and folders jobs

When a failover condition has been met, failover will be triggered automatically if you disabled the wait for user option during your failover configuration. If the wait for user before failover option is enabled, you will be notified in the console when a failover condition has been met. At that time, you will need to trigger it manually from the console when you are ready.

If you have paused your target, failover will not start if configured for automatic failover, and it cannot be initiated if configured for manual intervention. You must resume the target before failover will automatically start or before you can manually start it.

1. On the **Jobs** page, highlight the job that you want to failover and click **Failover or Cutover** in the toolbar.
2. Select the type of failover to perform.
   - **Failover to live data**—Select this option to initiate a full, live failover using the current data on the target. The target will stand in for the source by assuming the network identity of the failed source. User and application requests destined for the source server or its IP addresses are routed to the target.
   - **Perform test failover**—This option is not available for files and folders jobs.
   - **Failover to a snapshot**—This option is not applicable to files and folders jobs.
3. Select how you want to handle the data in the target queue.
   - **Apply data in target queues before failover or cutover**—All of the data in the target queue will be applied before failover begins. The advantage to this option is that all of the data that the target has received will be applied before failover begins. The disadvantage to this option is depending on the amount of data in queue, the amount of time to apply all of the data could be lengthy.
   - **Discard data in the target queues and failover or cutover immediately**—All of the data in the target queue will be discarded and failover will begin immediately. The advantage to this option is that failover will occur immediately. The disadvantage is that any data in the target queue will be lost.
4. When you are ready to begin failover, click **Failover**.

Failback and restoration for files and folders jobs

Failover occurred because the target was monitoring the source for a failure, and when a failure occurred, the target stood in for the source. User and application requests that were directed to the failed source are routed to the target.

While the users are accessing their data on the target, you can repair the issue(s) on the source. Before users can access the source again, you will need to restore the data from the target back to the source and perform failback. Failback is the process where the target releases the source identity it assumed during failover. Once failback is complete, user and application requests are no longer routed to the target, but back to the source.

Ideally, you want to restore your data from the target back to the source before you failback. This allows users who are currently accessing their data on the target because of failover to continue accessing their data. Restoration before failback reduces user downtime. Another method, which may be easier in some environments that have strict IP addressing policies, allows you to failback first and then restore the data from the target to the source. A possible disadvantage to this process is that users may experience longer downtime, depending on the amount of data to be restored, because they will be unable to access their data during both the restoration and the failback.

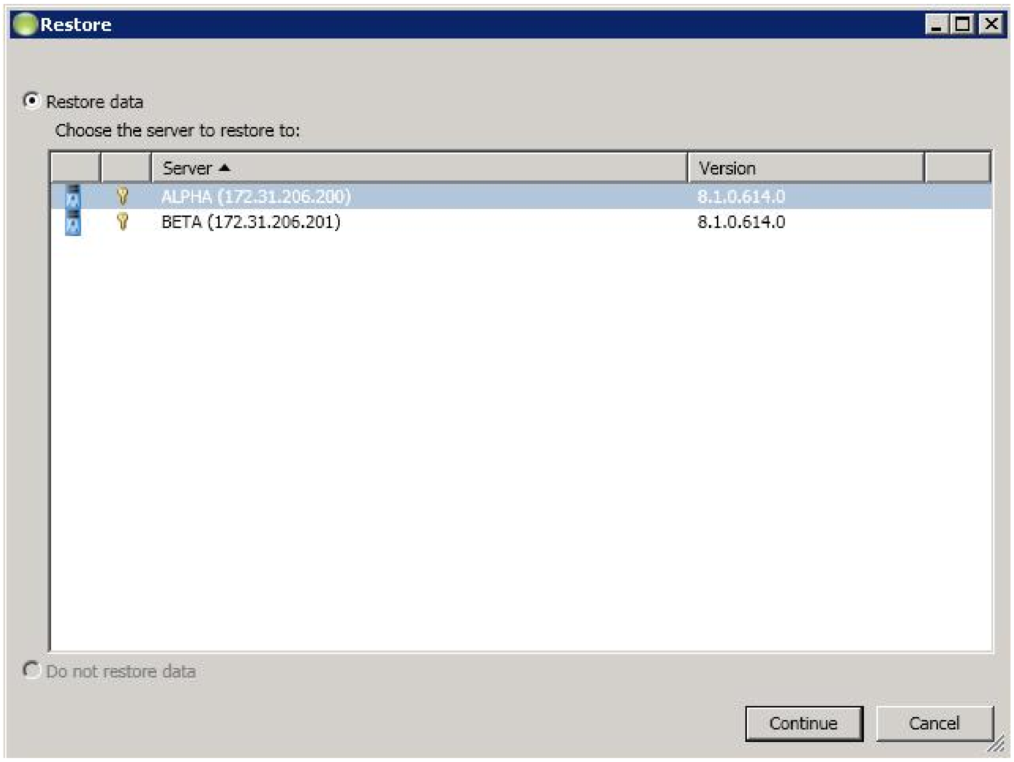
- *Restoring then failing back files and folders jobs* on page 101
- *Failing back then restoring files and folders jobs* on page 103

If you want to failback only, without performing a restoration, follow the instructions for failing back then restoring and you will be able to skip the restoration process. Keep in mind that if you skip the restoration process, any data changes that were made on the target during failover will be lost.

Restoring then failing back files and folders jobs

Restoring before failing back allows your users to continue accessing their data on the failed over target, which is standing in for the source, while you perform the restoration process. The key to this process is to keep the users off of the source, but allow the source and target to communicate to perform the restoration.

1. Resolve the problem(s) on the source that caused it to fail. Make sure in resolving the problems, that you do not bring the source on the network at this time because the target currently has the source's identity because of failover.
2. Disable all of the IP addresses on the source that you failed over to the target.
3. If you failed over all of your source IP addresses, change an existing IP address on the source to a new, unique IP address that the target can access. If you inserted your source server into the console using a reserved IP address or a public NAT address when you created the job, and you did not failover that IP address, you can skip this step.
4. Configure your new, unique IP address that you created or the reserved IP address that you are using so that it does not automatically register with DNS.
5. Bring the source onto the network using the IP address that the target can access. You can disregard any identity conflict errors.
6. Stop any applications that may be running on the source. Files must be closed on the source so that updated files from the target will overwrite the files on the source.
7. If you had to create a new, unique IP address on the source, you will have to remove the original source in the console and add it back in using the new, unique IP address. Use a local account, not a domain account, that is a member of the dtadmin group. Complete this step on the **Servers** page. If you inserted your source server into the console using a private IP address or a public NAT address when you created the job, you can skip this step.
8. On the **Jobs** page, highlight the job and click **Restore**.
9. Confirm **Restore data** is selected, then highlight your source server in the server list. If your server has a public NAT address, you can disable the default communication port and specify another port number to use, allowing the servers to communicate through a router.



● Restore data

Choose the server to restore to:

		Server ▲	Version	
		ALPHA (172.31.206.200)	8.1.0.614.0	
		BETA (172.31.206.201)	8.1.0.614.0	

○ Do not restore data

10. Click **Continue** to start the restoration.
11. When the restoration is complete, highlight the job and click **Failback**.
12. In the dialog box, highlight the job that you want to failback and click **Failback**.

> If you do not see your job after failback, remove and re-add your target to the console. The job may not be visible depending on where you are running the console from.

13. After failback is complete and the job is stopped, enable or add the IP addresses on the source that you disabled earlier. Make sure that you keep any new addresses that you created because the job is still using that address.
14. If you restored to a new source and are going to enable protection again, edit the job to reconfigure your failover settings.
15. Click **Start** to restart protection.

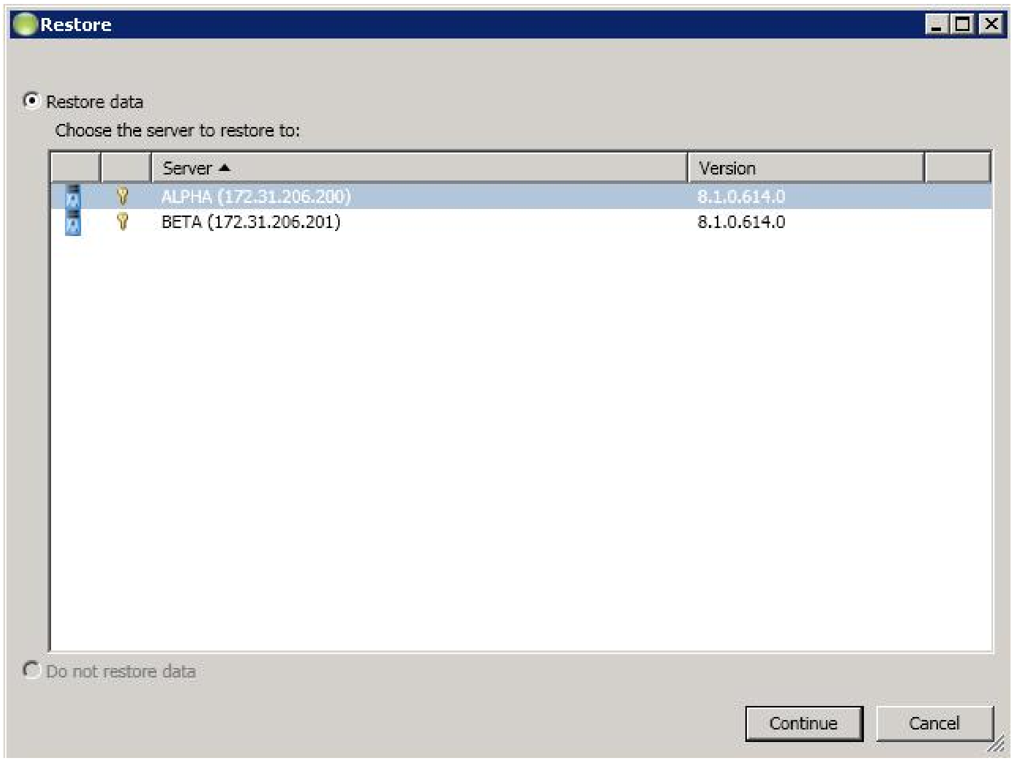
Failing back then restoring files and folders jobs

Failback before restoration can be a simpler process, but it may require additional downtime. The amount of downtime will depend on the amount of data to be restored. Users must be kept off of the source and target during this entire process.

1. Remove the source from the network and fix the issue that caused your source server to fail. Make sure in resolving the problems that you do not bring the source on the network at this time because the target currently has the source's identity because of failover.
2. Schedule a time for the failback and restoration process. Select a time that will have minimal disruption on your users.
3. When you are ready to begin the failback process, power on the source, if it is not on already. Make sure you do not connect it to the network. You must prohibit user access to both the source and target.
4. On the **Jobs** page, highlight the job and click **Failback**.
5. Highlight the job that you want to failback and click **Failback**.
6. Once failback is complete, connect the source to the network. Make sure that end users continue to be prohibited from accessing both the source and target because the updated data from the target needs to be restored back to the source.

   Depending on where you are running the console from, you may need to add the target back to the console after failback in order to see your job.

7. Stop any applications that may be running on the source. Files must be closed on the source so that updated files from the target will overwrite the files on the source.
8. On the **Jobs** page, highlight the job and click **Restore**.



● Restore data

Choose the server to restore to:

		Server ▲	Version
		ALPHA (172.31.206.200)	8.1.0.614.0
		BETA (172.31.206.201)	8.1.0.614.0

○ Do not restore data

9. If you want to skip the restoration, select **Do not restore data**, and click **Continue**. Keep in mind that if you skip the restoration process, any data changes that were made on the target during failover will be lost. If you want to restore the changed data from the target back to the source, select **Restore data**, highlight your source server in the server list. If your server has a public NAT address, you can disable the default communication port and specify another port number to use, allowing the servers to communicate through a router.
10. Click **Continue** to start the restoration.
11. When the restoration job is complete, the job will automatically be stopped. At this point, you can allow users to access the source again.
12. If you restored to a new source and are going to enable protection again, edit the job to reconfigure your failover settings.
13. Click **Start** to restart protection.

Chapter 5 Full server protection

Create a full server job when you want to protect the entire source, including the server's system state. You can also use it to protect an application server. This type of job is the most flexible, allowing you to go from physical to physical, physical to virtual, virtual to virtual, and virtual to physical. For full server protection, you will need to complete the following steps, in order.

1. Review the *Full server requirements* on page 106 to make sure your environment meets the requirements.
2. Install the Carbonite Replication Console on a Windows machine.
3. Install Carbonite Availability on your Linux source and target servers.
4. Add your servers to your Carbonite Replication Console. See *Adding servers* on page 34.
5. Create your Linux full server job. See *Creating a full server job* on page 114.

For installation and licensing instructions, see the *Carbonite Availability and Carbonite Migrate Installation, Licensing, and Activation* document.

Once your job is created and running, see the following sections to manage your job.

- *Managing and controlling full server jobs* on page 133—You can view status information about your job and learn how to control the job.
- *Failing over full server jobs* on page 150—Use this section when a failover condition has been met or whenever you want to failover.
- *Reversing full server jobs* on page 152—Use this section to reverse protection. The source (what was your original target hardware) is now sending data to the target (what was your original source hardware).

Full server requirements

Use these requirements for Linux full server protection. Keep in mind that a target server may meet these requirements but may not be suitable to stand-in for a source in the event of a source failure. See *Target compatibility* on page 112 for additional information regarding an appropriate target server for your particular source.

- **Source and target servers**—The source and target servers can be a physical or virtual server running any of the following operating systems.
  - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
    - **Version**—5.9 through 5.11
    - **Kernel type for x86 (32-bit) architectures**—Default, SMP, Xen, PAE
    - **Kernel type for x86-64 (64-bit) architectures**—Default, SMP, Xen
    - **File system**—Ext3, Ext4, XFS
    - **Notes**—Oracle Enterprise Linux support is for the mainline kernel only, not the Unbreakable kernel.
  - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
    - **Version**—6.8 through 6.10
    - **Kernel type for x86 (32-bit) architectures**—Default
    - **Kernel type for x86-64 (64-bit) architectures**—Default
    - **File system**—Ext3, Ext4, XFS (64-bit only)
  - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
    - **Version**—7.3 through 7.5
    - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
    - **Kernel type for x86-64 (64-bit) architectures**—Default
    - **File system**—Ext3, Ext4, XFS
    - **Notes**—For full server jobs, if your source is running version 7.3, your target must be running version 7.3 also. Because of operating system changes, version 7.3 on the source cannot be used with 7.4 or later on the target.
  - **Operating system**—SUSE Linux Enterprise
    - **Version**—11.2 through 11.4
    - **Kernel type for x86 (32-bit) architectures**—Default, Xen, XenPAE, VMI
    - **Kernel type for x86-64 (64-bit) architectures**—Default, Xen
    - **File system**—Ext3, XFS
  - **Operating system**—SUSE Linux Enterprise
    - **Version**—12.1 through 12.3
    - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
    - **Kernel type for x86-64 (64-bit) architectures**—Default
    - **File system**—Ext3, Ext4, XFS, Btrfs
    - **Notes**—If you are planning to convert an existing file system to Btrfs, you must delete any existing Carbonite Availability jobs and re-create them after converting to

Btrfs. Also Btrfs cannot be failed over together with ext4. Btrfs and ext4 can be combined with other file systems but not with each other.

- **Operating system**—Ubuntu
  - **Version**—12.04.3, 12.04.4, and 12.04.5
  - **Kernel type for x86 (32-bit) architectures**—Generic
  - **Kernel type for x86-64 (64-bit) architectures**—Generic
  - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
  - **Version**—14.04.3, 14.04.4, and 14.04.5
  - **Kernel type for x86 (32-bit) architectures**—Generic
  - **Kernel type for x86-64 (64-bit) architectures**—Generic
  - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
  - **Version**—16.04.2, 16.04.3, and 16.04.4
  - **Kernel type for x86 (32-bit) architectures**—Generic
  - **Kernel type for x86-64 (64-bit) architectures**—Generic
  - **File system**—Ext2, Ext3, Ext4, XFS
- **Operating system**—Ubuntu
  - **Version**—18.04.0
  - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
  - **Kernel type for x86-64 (64-bit) architectures**—Generic
  - **File system**—Ext2, Ext3, Ext4, XFS

For all operating systems except Ubuntu, the kernel version must match the expected kernel for the specified release version. For example, if /etc/redhat-release declares the system to be a Redhat 7.5 system, the kernel that is installed must match that.

Carbonite Availability does not support stacking filesystems, like eCryptFS.

If Carbonite Availability does not have the driver binary files for the kernel you are using, they can be compiled automatically, but you need the build-essential package for them to be installed. Run apt-get install build-essential to install the build tools and then restart the DT service. This will build the driver from the source and load it.

- **Packages and services**—Each Linux server must have the following packages and services installed before you can install and use Carbonite Availability. See your operating system documentation for details on these packages and utilities.
  - sshd (or the package that installs sshd)
  - lsb
  - parted
  - dmidecode

    - scp
    - which
- **Source and target preparation**—Make sure your source and target servers are prepared for mirroring, replication, and failover by following these guidelines.
    - Uninstall any applications or operating system features that are not needed from both your source and target. Ideally, your target should be as clean and simple a configuration as possible.
    - Install on the source any drivers that are required on the target after failover. For example, you need to install on the source any NIC drivers that will be required on the target after failover.
    - Resolve any maintenance updates on the source that may require the server to be rebooted before failover.
    - Do not failover if the target is waiting on a reboot after applying maintenance. If failover occurs before the required reboot, the target may not operate properly or it may not boot.
- **System memory**—The minimum system memory on each server is 1 GB.
- **Disk space for program files**—This is the amount of disk space needed for the Carbonite Availability program files. This is approximately 400 MB on each Linux server.

    Make sure you have additional disk space for Carbonite Availability queuing, logging, and so on.

- **Server name**—Carbonite Availability includes Unicode file system support, but your server name must still be in ASCII format. Additionally, all Carbonite Availability servers must have a unique server name.
- **Protocols and networking**—Your servers must meet the following protocol and networking requirements.
    - Your servers must have TCP/IP with static IP addressing.
    - IPv4 is the only supported version.
    - If you are using Carbonite Availability over a WAN and do not have DNS name resolution, you will need to add the host names to the local hosts file on each server running Carbonite Availability.
    - Ubuntu Netplan is not a supported configuration. You must be running NetworkManager natively without Netplan. If you protect and failover a Linux server with Netplan, you will have to configure networking manually after failover.
- **NAT support**—Carbonite Availability supports NAT environments with the following caveats.
    - Only IPv4 is supported.
    - Only standalone servers are supported.
    - Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.

- DNS failover and updates will depend on your configuration
    - Only the source or target can be behind a router, not both.
    - The DNS server must be routable from the target
- **Name resolution**—Your servers must have name resolution or DNS. The Carbonite Replication Console must be able to resolve the target, and the target must be able to resolve all source servers. For details on name resolution options, see your Linux documentation or online Linux resources.
- **Ports**—Port 1501 is used for localhost communication between the engine and management service and should be opened inbound and outbound for both TCP and UDP in iptables. Ports 1500, 1505, 1506, 6325, and 6326 are used for component communication and must be opened inbound and outbound for both TCP and UDP on any firewall that might be in use.
- **Security**—Carbonite Availability security is granted through membership in user groups. The groups can be local or LDAP (Lightweight Directory Access Protocol). A user must provide a valid local account that is a member of the Carbonite Availability security groups.
- **SELinux policy**—The SELinux configuration should match on the source and target. For example, if the SELinux configuration is permissive on the source, it should be permissive on the target.
- **UEFI, trusted boot, secure boot**—UEFI (Unified Extensible Firmware Interface) is supported on the source and target, however, trusted boot (tboot), secure boot, or other volume blocking mechanisms are not supported on the source and target.

If you are using SUSE Linux Enterprise version 11.4, you cannot mix UEFI and BIOS. With this version, the source and target must be the same.

- **Docker**—Your source cannot be a Docker host.
- **Mount option**—The mount option noexec is not supported on the /tmp filesystem.
- **Kernel**—Paravirtualized kernels are not supported on the source and target.
- **VMware Tools**—Any VMWare guest running Carbonite Availability should have the appropriate VMWare Tools package installed.
- **Snapshots**—You can take and failover to snapshots using a full server job. Keep in mind the following caveats.
    - You must have LVM on your source and target server. The snapshots will be stored in the LVM volume group within the same volume group as the parent volume, so make sure you have enough free space to accommodate the snapshots. If you do not have enough free space, Carbonite Availability will delete enough snapshots (typically one) to free space for the new snapshot. Bad or overflow snapshots will be deleted first and if there are none of those, then the oldest snapshot will be deleted.
    - There may be a performance impact if you take a lot of snapshots. The more snapshots you have, the longer it may take to create a new snapshot because the volume write time slows down for every snapshot because every block written could cause a write to each snapshot volume.
    - Snapshots are not supported with Btrfs file systems.

  - Snapshots are not supported on Ubuntu 12.04.x. Once a job is created, you will still see Carbonite Availability functionality to take snapshots, but due to operating system limitations specifically with Ubuntu 12.04.x, the snapshots will not be usable.
- **Test failover**—Test failover allows you to keep your job intact and use a third machine to test the failover process. To complete the test functionality, Carbonite Availability use LVM snapshots. Keep in mind the following for using test failover.
  - Your source must have / or /opt/dbtk/var/lib on LVM in order to use the test failover feature.
  - All data volumes must be under LVM for test failover.
  - Test failover is not supported for Btrfs file systems.
  - Your test failover server must have the same volume configuration (BIOS or UEFI) as your target server.
  - The source, target, and protection job will remain online and uninterrupted during the test.
  - During the test, any scheduled snapshots for the protection job will be deferred until after the test. Manual snapshots will be disabled until after the test.
  - The test will be performed using the test failover settings configured during job creation.
  - The test failover will take a snapshot of the current data on the target and mirror the data from the snapshot to the test failover machine using the same mirroring options as the protection job.
  - Once the mirror is complete, the test failover machine is rebooted automatically to finalize the test failover process.
  - The test failover machine will maintain its own networking which keeps it isolated from the rest of the network in order to avoid network conflicts and redirecting clients. Applications or functionality that relies on the source networking may not be fully testable with the test machine networking.
  - When you are finished with your test, undo it.
  - When you undo a test failover, the snapshot will be deleted.
  - At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)
- **Supported configurations**—The following table identifies the supported configurations for a full server job.

Server Configuration	Description	Supported	Not Supported
One to one active/standby	You can protect a single source to a single target. The target has no production activity. The source is the only server actively replicating data.	X	
One to one active/active	You cannot protect a single source to a single target where each server acts as both a source and target actively replicating data to each other.		X

Server Configuration	Description	Supported	Not Supported
Many to one	You cannot protect many source servers to one target server.		X
One to many	You can protect a single source to multiple target servers. The source is the only server actively replicating data. This will create redundant copies of your source. You will only be able to configure reverse protection for the first job. Subsequent jobs from that source will have reverse protection disabled.	X	
Chained	You cannot protect a single source to a single target, where the target then acts as a source, sending the same data from the original source to a final target server.		X
Single server	You cannot protect a single source to itself.		X
Standalone to standalone	Your servers can be in a standalone to standalone configuration.	X	
Standalone to cluster	Your servers cannot be in a standalone to cluster configuration.		X
Cluster to standalone	Your servers cannot be in a cluster to standalone configuration.		X
Cluster to cluster	Your servers cannot be in a cluster to cluster configuration.		X

Target compatibility

- **Operating system version**—The source and target must have the same distribution and major version. For example, you cannot have a Red Hat version 5.8 source failing over to a Red Hat version 6.4 target. The two servers do not have to have the same minor version. For example, you can failover Red Hat version 6.4 to Red Hat version 6.5.
- **Source and target preparation**—Make sure your source and target servers are prepared for mirroring, replication, and failover by following these guidelines.
    - Uninstall any applications or operating system features that are not needed from both your source and target. Ideally, your target should be as clean and simple a configuration as possible.
    - Install on the source any drivers that are required on the target after failover. For example, you need to install on the source any NIC drivers that will be required on the target after failover.
    - Resolve any maintenance updates on the source that may require the server to be rebooted before failover.
    - Do not failover if the target is waiting on a reboot after applying maintenance. If failover occurs before the required reboot, the target may not operate properly or it may not boot.
- **Architecture**—The source and the target must have the same architecture. For example, you cannot failover a 32-bit server to a 64-bit server.
- **Processors**—There are no limits on the number or speed of the processors, but the source and the target should have at least the same number of processors. If the target has fewer processors or slower speeds than the source, there will be performance impacts for the users after failover.
- **Memory**—The target memory should be within 25% (plus or minus) of the source. If the target has much less memory than the source, there will be performance impacts for the users after failover.
- **Network adapters**—You must map at least one NIC from the source to one NIC on the target. If you have NICs on the source that are not being used, it is best to disable them. If the source has more NICs than the target, some of the source NICs will not be mapped to the target. Therefore, the IP addresses associated with those NICs will not be available after failover. If there are more NICs on the target than the source, the additional NICs will still be available after failover and will retain their pre-failover network settings.
- **File system format**—The source and the target must have the file system format on each server. For example, if you have Ext3 on the source, you cannot have XFS on the target. In that case, the target must also be Ext3.
- **Volumes**—There are no limits to the number of volumes you can protect on the source, although you are bound by operating system limits.

  For each non-system volume you are protecting on the source, the target must have a matching volume. For example, if you are protecting /data and /home on the source, the target must also have /data and /home. Additional target volumes are preserved and available after failover with all data still accessible, however you will be unable to reverse protection if the target has more volumes than the source.

  The system volumes / and /boot do not have this matching volume limitation. If you have / and /boot on different volumes on the source, they can exist on a single volume on the target. If you have / and /boot on the same volume on the source, they can exist on different volumes on the target.

- **Carbonite Availability version**—If you will be using the reverse feature with your full server job, your source and target must be running the same Carbonite Availability version.
- **Disk space**—The target must have enough space to store the data from the source. This amount of disk space will depend on the applications and data files you are protecting. The more data you are protecting, the more disk space you will need. The target must also have enough space to store, process, and apply the source's system state data. If you will be enabling reverse protection, the source must have enough space to store, process, and apply the target's system state data.

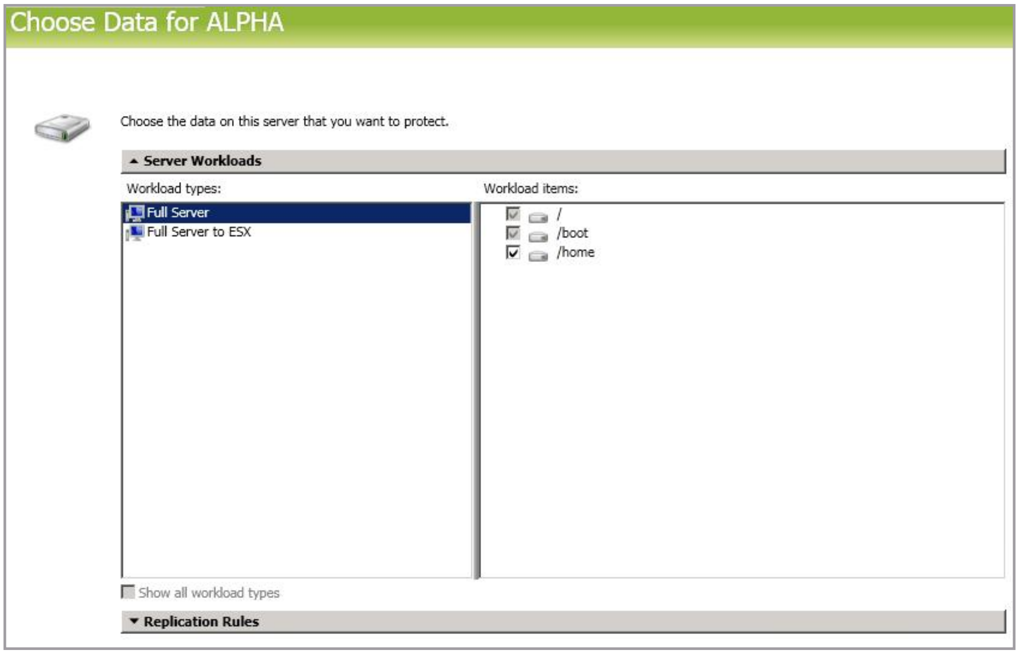
  A copy of the source data and system state will be staged on the target in a /dtstaging location for each mount point. For example, / will be staged in /dtstaging and /boot will be staged in /boot/dtstaging. For reverse protection, the same staging structure is used. You can predict how much space you will need in the staging folders by the amount of used space on the source or target, respectively.

  Keep in mind you should have extra space available on each server for any data growth.
- **Services**—Ideally, you should have the same services and run levels on the source and target.

Creating a full server job

Use these instructions to create a full server job.

1. From the **Servers** page, right-click the server you want to protect and select **Protect**. You can also highlight a server, click **Create a New Job** in the toolbar, then select **Protect**.
2. Choose the type of workload that you want to protect. Under **Server Workloads**, in the **Workload types** pane, select **Full Server**. In the **Workload items** pane, select the volumes on the source that you want to protect.



Choose Data for ALPHA

Choose the data on this server that you want to protect.

Server Workloads

Workload types:

- Full Server
- Full Server to ESX

Workload items:

- ☑ /
- ☑ /boot
- ☑ /home

☐ Show all workload types

Replication Rules

Unsupported file systems will be displayed but will not be accessible.

3. By default, Carbonite Availability selects the system and boot volumes for protection. You will be unable to deselect these volumes. Select any other volumes on the source that you want to protect.

   If desired, click the **Replication Rules** heading and expand the volumes under **Folders**. You will see that Carbonite Availability automatically excludes particular files that cannot be used during the protection. If desired, you can exclude other files that you do not want to protect, but be careful when excluding data. Excluded volumes, folders, and/or files may compromise the integrity of your installed applications.

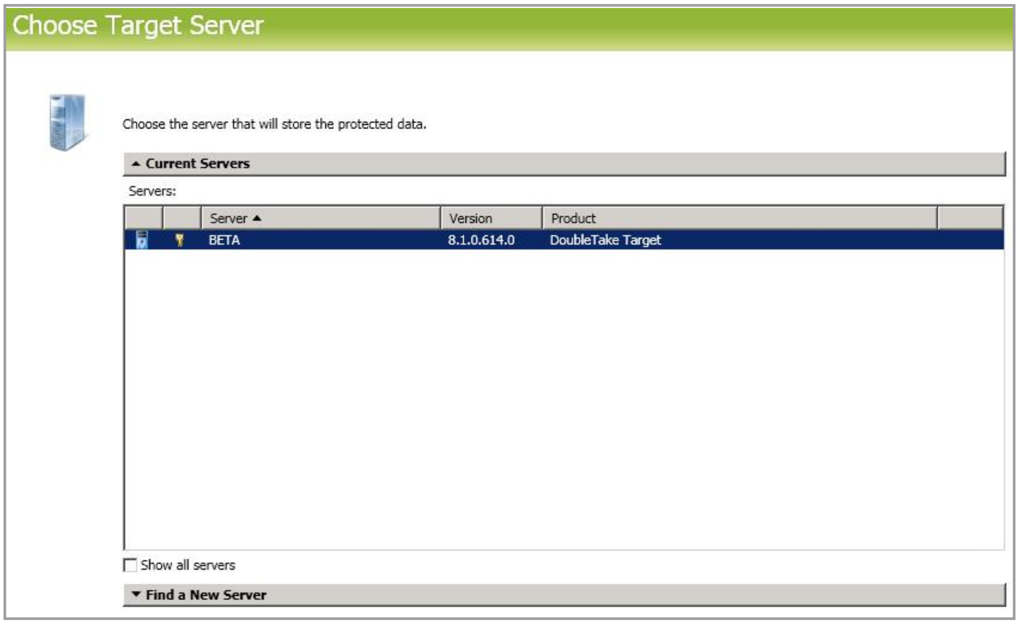
   Volumes and folders with a green highlight are included completely. Volumes and folders highlighted in light yellow are included partially, with individual files or folders included. If there is no highlight, no part of the volume or folder is included. To modify the items selected, highlight a volume, folder, or file and click **Add Rule**. Specify if you want to **Include** or **Exclude** the item. Also, specify if you want the rule to be recursive, which indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select **Recursive**, the rule will not be applied to subdirectories.

You can also enter wildcard rules, however you should do so carefully. Rules are applied to files that are closest in the directory tree to them. If you have rules that include multiple folders, an exclusion rule with a wild card will need to be added for each folder that it needs applied to. For example, if you want to exclude all .log files from /home and your rules include /home, /home/folder1, and /home/folder2, you would need to add the exclusion rule for the root and each subfolder rule. So you will need to add exclude rules for /home/*.log , /home/folder1/*.log, and /home/folder2/*.log.

If you need to remove a rule, highlight it in the list at the bottom and click **Remove Rule**. Be careful when removing rules. Carbonite Availability may create multiple rules when you are adding directories. For example, if you add /home/admin to be included in protection, then /home will be excluded. If you remove the /home exclusion rule, then the /home/admin rule will be removed also.

If you return to this page using the **Back** button in the job creation workflow, your **Workload Types** selection will be rebuilt, potentially overwriting any manual replication rules that you specified. If you do return to this page, confirm your **Workload Types** and **Replication Rules** are set to your desired settings before proceeding forward again.

4. Click **Next** to continue.
5. Choose your target server. This is the server that will store the replica data from the source, and in the event of a failover, it will become your source.



Choose Target Server

Choose the server that will store the protected data.

		Server ▲	Version	Product
		BETA	8.1.0.614.0	DoubleTake Target

- **Current Servers**—This list contains the servers currently available in your console session. Servers that are not licensed for the workflow you have selected and those not applicable to the workload type you have selected will be filtered out of the list. Select your target server from the list. If the server you are looking for is not displayed, enable **Show all servers**. The servers in red are not available for the source server or workload type you have selected. Hover your mouse over an unavailable server to see a reason why this server is unavailable.

- **Find a New Server**—If the server you need is not in the **Current Servers** list, click the **Find a New Server** heading. From here, you can specify a server along with credentials for logging in to the server. If necessary, you can click **Browse** to select a server from a network drill-down list.

If you enter the target server's fully-qualified domain name, the Carbonite Replication Console will resolve the entry to the server short name. If that short name resides in two different domains, this could result in name resolution issues. In this case, enter the IP address of the server.

When specifying credentials for a new server, specify a user that is a member of the local dtadmin security group.

6. Click **Next** to continue.

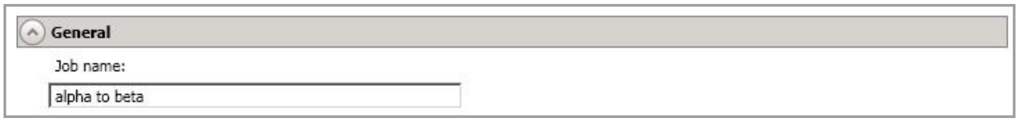
You may be prompted for a route from the target to the source. This route is used so the target can communicate with the source to build job options. This dialog box will be displayed only if needed.

7. You have many options available for your Linux full server job. Configure those options that are applicable to your environment.

   Go to each page identified below to see the options available for that section of the **Set Options** page. After you have configured your options, continue with the next step on page 131.

   - *General* on page 117
   - *Failover Monitor* on page 118
   - *Test Failover* on page 120
   - *Failover Options* on page 122
   - *Failover Identity* on page 123
   - *Reverse Protection and Routing* on page 124
   - *Network Adapter Options* on page 126
   - *Mirror, Verify & Orphaned Files* on page 127
   - *Snapshots* on page 129
   - *Compression* on page 130
   - *Bandwidth* on page 131

General

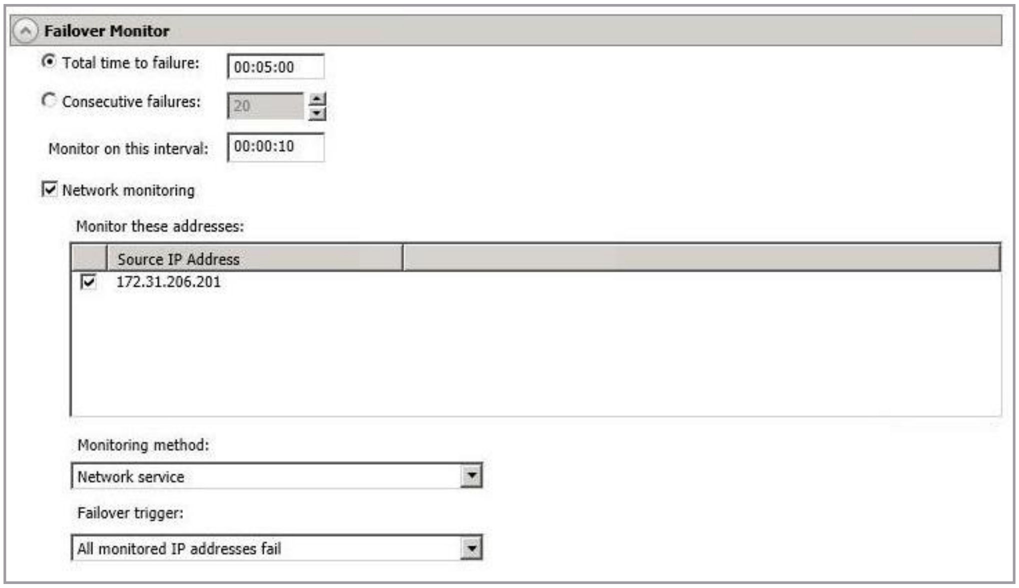


Job name:

alpha to beta

For the **Job name**, specify a unique name for your job.

Failover Monitor



Failover Monitor

● Total time to failure: 00:05:00

○ Consecutive failures: 20

Monitor on this interval: 00:00:10

☑ Network monitoring

Monitor these addresses:

	Source IP Address
☑	172.31.206.201

Monitoring method: Network service

Failover trigger: All monitored IP addresses fail

- **Total time to failure**—Specify, in hours:minutes:seconds, how long the target will keep trying to contact the source before the source is considered failed. This time is precise. If the total time has expired without a successful response from the source, this will be considered a failure.

  Consider a shorter amount of time for servers, such as a web server or order processing database, which must remain available and responsive at all times. Shorter times should be used where redundant interfaces and high-speed, reliable network links are available to prevent the false detection of failure. If the hardware does not support reliable communications, shorter times can lead to premature failover. Consider a longer amount of time for machines on slower networks or on a server that is not transaction critical. For example, failover would not be necessary in the case of a server restart.

- **Consecutive failures—**Specify how many attempts the target will make to contact the source before the source is considered failed. For example, if you have this option set to 20, and your source fails to respond to the target 20 times in a row , this will be considered a failure.
- **Monitor on this interval**—Specify, in hours:minutes:seconds, how long to wait between attempts to contact the source to confirm it is online. This means that after a response (success or failure) is received from the source, Carbonite Availability will wait the specified interval time before contacting the source again. If you set the interval to 00:00:00, then a new check will be initiated immediately after the response is received.

  If you choose **Total time to failure**, do not specify a longer interval than failure time or your server will be considered failed during the interval period.

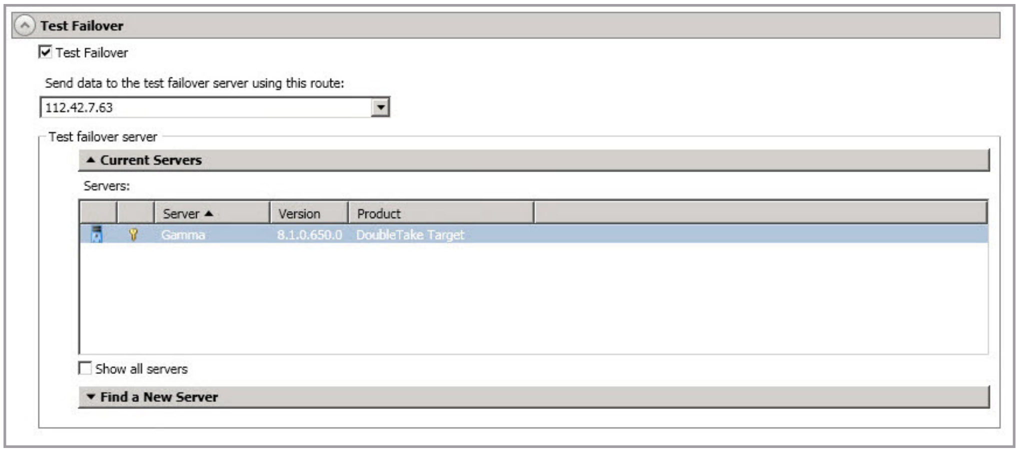
  If you choose **Consecutive failures**, your failure time is calculated by the length of time it takes your source to respond plus the interval time between each response, times the number of consecutive failures that can be allowed. That would be (response time + interval) * failure number. Keep in mind that timeouts from a failed check are included in the response time, so your failure time will not be precise.

- **Network monitoring**—With this option, the target will monitor the source using a network ping.
    - **Monitor these addresses**—Select each **Source IP Address** that you want the target to monitor. If you are using a NAT environment, do not select a private IP address on the source because the target cannot reach the source's private address, thus causing an immediate failure.
    - **Monitoring method—**This option determines the type of network ping used for failover monitoring.
        - **Network service**—Source availability will be tested by an ICMP ping to confirm the route is active.
        - **Replication service**—Source availability will be tested by a UDP ping to confirm the Double-Take service is active.
        - **Network and replication services—**Source availability will be tested by both an ICMP ping to confirm the route is active and a UDP ping to confirm the Double-Take service is active. If either monitoring method fails, failover will be triggered.
    - **Failover trigger—**If you are monitoring multiple IP addresses, specify when you want a failover condition to be triggered.
        - **One monitored IP address fails—**A failover condition will be triggered when any one of the monitored IP addresses fails. If each IP address is on a different subnet, you may want to trigger failover after one fails.
        - **All monitored IP addresses fail**—A failover condition will be triggered when all monitored IP addresses fail. If there are multiple, redundant paths to a server, losing one probably means an isolated network problem and you should wait for all IP addresses to fail.

Test Failover

These options allow you to perform a test failover. Keep in mind the following for using test failover.

- Your source must have / or /opt/dbtk/var/lib on LVM in order to use the test failover feature.
- All data volumes must be under LVM for test failover.
- Test failover is not supported for Btrfs file systems.
- Your test failover server must have the same volume configuration (BIOS or UEFI) as your target server.
- The source, target, and protection job will remain online and uninterrupted during the test.
- During the test, any scheduled snapshots for the protection job will be deferred until after the test. Manual snapshots will be disabled until after the test.
- The test will be performed using the test failover settings configured during job creation.
- The test failover will take a snapshot of the current data on the target and mirror the data from the snapshot to the test failover machine using the same mirroring options as the protection job.
- Once the mirror is complete, the test failover machine is rebooted automatically to finalize the test failover process.
- The test failover machine will maintain its own networking which keeps it isolated from the rest of the network in order to avoid network conflicts and redirecting clients. Applications or functionality that relies on the source networking may not be fully testable with the test machine networking.
- When you are finished with your test, undo it.
- When you undo a test failover, the snapshot will be deleted.
- At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)



Test Failover

☑ Test Failover

Send data to the test failover server using this route: 112.42.7.63

Test failover server

▲ Current Servers

Servers:

		Server ▲	Version	Product
		Gamma	8.1.0.650.0	DoubleTake Target

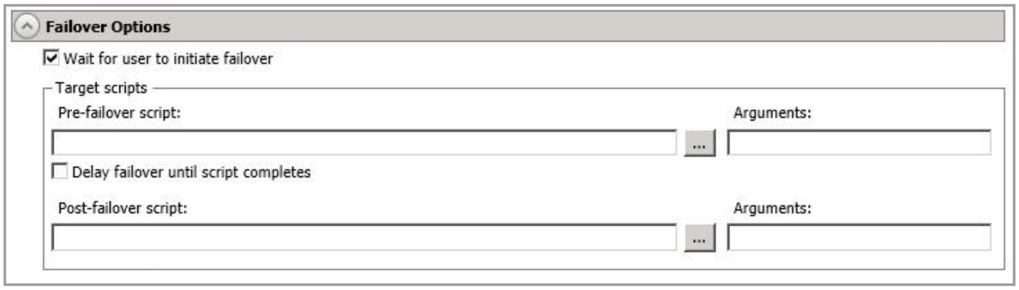
□ Show all servers

▼ Find a New Server

- **Test Failover**—Enable this option to be able to perform test failover.
- **Send data to the test failover server using this route**—Select or enter a route to use on the test failover server for mirroring the data from the snapshot to the test failover server.

- **Test failover server**—Select the server you want to use for the test failover.
    - **Current Servers—**This list contains the servers currently available in your console session. Servers that are not applicable to test failover will be filtered out of the list. Select your test failover server from the list. If the server you are looking for is not displayed, enable **Show all servers**. The servers in red are not available. Hover your mouse over an unavailable server to see a reason why this server is unavailable.
    - **Find a New Server—**If the server you need is not in the **Current Servers** list, click the **Find a New Server** heading. From here, you can specify a server along with credentials for logging in to the server. If necessary, you can click **Browse** to select a server from a network drill-down list. After you have identified or located the server, click **Add Server**. If there are any issues connecting to that server, you will see an error in yellow at the top of the page. If there are no issues, you can continue.

Failover Options



Failover Options

☑ Wait for user to initiate failover

Target scripts

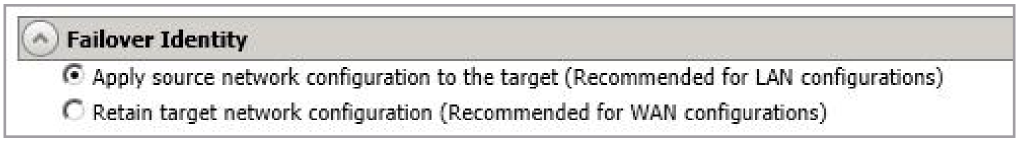
Pre-failover script: ... Arguments:

□ Delay failover until script completes

Post-failover script: ... Arguments:

- **Wait for user to initiate failover**—The failover process can wait for you to initiate it, allowing you to control when failover occurs. When a failure occurs, the job will wait in **Failover Condition Met** for you to manually initiate the failover process. Disable this option if you want failover to occur immediately when a failure occurs.
- **Target Scripts**—You can customize failover by running scripts on the target. Scripts may contain any valid Linux command, executable, or shell script file. The scripts are processed using the same account running the Double-Take Management service. Examples of functions specified in scripts include stopping services on the target before failover because they may not be necessary, stopping services on the target that need to be restarted with the source’s machine name and/or IP address, starting services or loading applications that are in an idle, standby mode waiting for failover to occur, notifying the administrator before and after failover occurs, and so on. There are two types of failover scripts.
  - **Pre-failover script**—This script runs on the target at the beginning of the failover process. Specify the full path and name of the script file.
  - **Delay until script completes**—Enable this option if you want to delay the failover process until the associated script has completed. If you select this option, make sure your script handles errors, otherwise the failover process may never complete if the process is waiting on a script that cannot complete.
  - **Post-failover script—**This script runs on the recovered source at the end of the failover process. Specify the full path and name of the script file.
  - **Arguments—**Specify a comma-separated list of valid arguments required to execute the script.

Failover Identity



Failover Identity

● Apply source network configuration to the target (Recommended for LAN configurations)

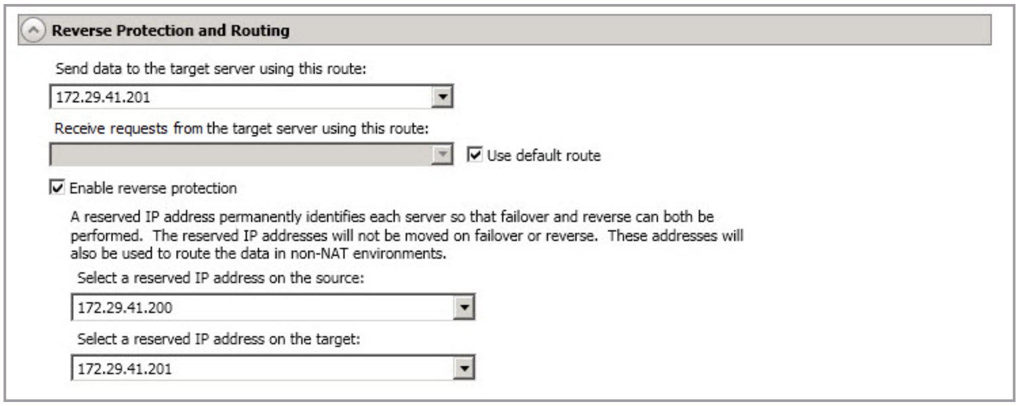
○ Retain target network configuration (Recommended for WAN configurations)

- **Apply source network configuration to the target**—If you select this option, your source IP addresses will failover to the target. If your target is on the same subnet as the source (typical of a LAN environment), you should select this option. Do not select this option if you are using a NAT environment that has a different subnet on the other side of the router.

  Do not apply the source network configuration to the target in a WAN environment unless you have a VPN infrastructure so that the source and target can be on the same subnet, in which case IP address failover will work the same as a LAN configuration. If you do not have a VPN, you will have to reconfigure the routers by moving the source's subnet from the source's physical network to the target's physical network. There are a number of issues to consider when designing a solution that requires router configuration to achieve IP address failover. Since the route to the source's subnet will be changed at failover, the source server must be the only system on that subnet, which in turn requires all server communications to pass through a router. Additionally, it may take several minutes or even hours for routing tables on other routers throughout the network to converge.

- **Retain target network configuration**—If you select this option, the target will retain all of its original IP addresses. If your target is on a different subnet (typical of a WAN or NAT environment), you should select this option.

Reverse Protection and Routing



Reverse Protection and Routing

Send data to the target server using this route:

172.29.41.201

Receive requests from the target server using this route:

☑ Use default route

☑ Enable reverse protection

A reserved IP address permanently identifies each server so that failover and reverse can both be performed. The reserved IP addresses will not be moved on failover or reverse. These addresses will also be used to route the data in non-NAT environments.

Select a reserved IP address on the source:

172.29.41.200

Select a reserved IP address on the target:

172.29.41.201

- **Send data to the target server using this route**—By default, Carbonite Availability will select a target route for transmissions. If desired, specify an alternate route on the target that the data will be transmitted through. This allows you to select a different route for Carbonite Availability traffic. For example, you can separate regular network traffic and Carbonite Availability traffic on a machine with multiple IP addresses. You can also select or manually enter a public IP address (which is the public IP address of the server's router) if you are using a NAT environment.
- **Receive requests from the target server using this route**—By default, Carbonite Availability will select a route from the target for command and status requests. If desired, specify an alternate route on the target that the commands will be transmitted from. This allows you to select a different route for Carbonite Availability management communication. You can also manually enter a public IP address (which is the public IP address of the server's router) if you are using a NAT environment.
- **Use default route**—Select this option to disable the drop-down list that allows you to select the route from the target server. When this option is enabled, the default route will automatically be used.
- **Enable reverse protection**—After failover, your target server is lost. Reverse protection allows you to store a copy of the target's system state on the source server, so that the target server will not be lost. The reverse process will bring the target identity back on the source hardware and establish protection. After the reverse, the source (running on the original target hardware) will be protected to the target (running on the original source hardware).

  If you do not use reverse protection, after a failover, your target server will be lost. In order to continue protecting your data, you will have to manually rebuild your original source and restart protection, which can be a long and complicated process. Also, if you disable reverse, you will lose the activated target license after failover.

  You may want to consider having two IP addresses on each server. This will allow you to monitor and failover one (or more) IP addresses, while still leaving an IP address that does not get failed over. This IP address that is not failed over is called a reserved IP address and can be used for the reverse process. The reserved IP address remains with the server hardware. Ideally, the reserved IP address should not be used for production communications. The reserved IP address can be on the same or a different subnet from your production IP addresses, however if the subnet is different, it should be on a different network adapter. The reserved IP addresses will also be used to route Carbonite Availability data.

You do not have to have a second IP address on each server. It is acceptable to use the production IP address for reverse protection, as long as you are selecting the option to retain the target configuration.

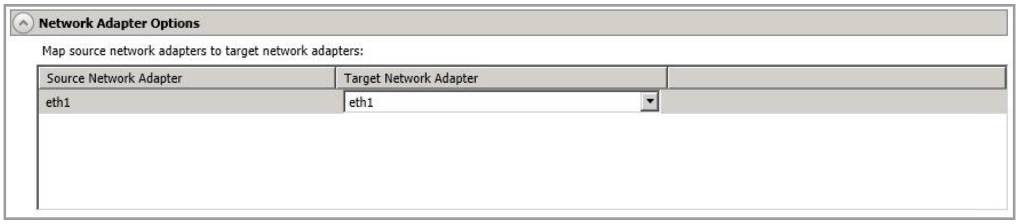
- **Select a reserved IP address on the source**—Specify an IP address on the source which will be used to permanently identify the source server. The IP address you specify will not be failed over to the target in the event of a failure. This allows you to reverse protection back to the source after a failover.
- **Send data to source after reverse using this route**—This field will only be displayed if the console recognizes that your source address is a public NAT address. In that case, you can specify the route.
- **Select a reserved IP address on the target**—Specify an IP address on the target which will be used to permanently identify the target server. The IP address you specify will not be lost during failover. This allows you to reverse protection back to the source after a failover. In a non-NAT environment, this address will override the target route above and be used to route the data to the target server.

When reverse protection is enabled, your source server must have space to store, process, and apply the target's system state data.

When the job is first started and reverse protection is enabled, an image of the target's system state is mirrored to the source server. This mirror may cause a performance impact on your source server. This impact is only temporary, and system performance will return to normal when the reverse protection mirror is complete.

To maintain system performance on the source, the target's system state is not continuously replicated to the source. You can manually update the image of the target's system state by viewing the job details and clicking **Update** under **Target Server Image**. See *Viewing full server job details* on page 141.

Network Adapter Options



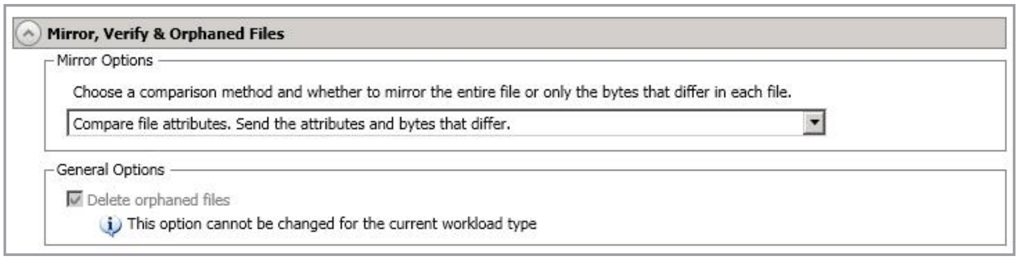
Network Adapter Options

Map source network adapters to target network adapters:

Source Network Adapter	Target Network Adapter
eth1	eth1

For **Map source network adapters to target network adapters**, specify how you want the IP addresses associated with each NIC on the source to be mapped to a NIC on the target. Do not mix public and private networks. Also, if you have enabled reverse protection, make sure that your NICs with your reserved IP addresses are mapped to each other.

Mirror, Verify & Orphaned Files



Mirror, Verify & Orphaned Files

Mirror Options

Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.

Compare file attributes. Send the attributes and bytes that differ.

General Options

☑ Delete orphaned files

This option cannot be changed for the current workload type

- **Mirror Options**—Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.
    - **Do not compare files. Send the entire file.**—Carbonite Availability will not perform any comparisons between the files on the source and target. All files will be mirrored to the target, sending the entire file. This option requires no time for comparison, but it can be slower because it sends the entire file. However, it is useful for configurations that have large data sets with millions of small files that are frequently changing and it is more efficient to send the entire file. You may also need to use this option if configuration management policies require sending the entire file.
    - **Compare file attributes. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and will mirror only the attributes and bytes that are different. This option is the fastest comparison method and fastest mirror option. Files that have not changed can be easily skipped. Also files that are open and require a checksum mirror can be compared.
    - **Compare file attributes and data. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and the file data and will mirror only the attributes and bytes that are different. This comparison method is not as fast because every file is compared, regardless of whether the file has changed or is open. However, sending only the attributes and bytes that differ is the fasted mirror option.

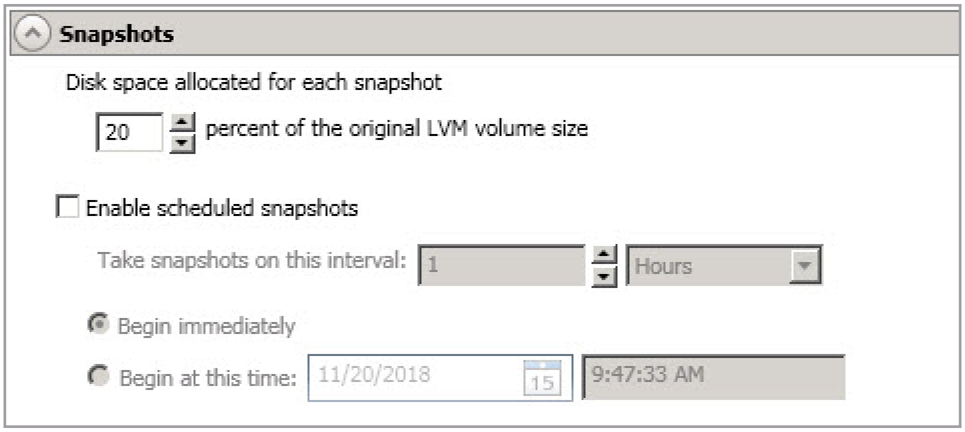
  If a file is small enough that mirroring the entire file is faster than comparing it and then mirroring it, Carbonite Availability will automatically mirror the entire file.

- **General Options**—Choose your general mirroring options.
    - **Delete orphaned files**—An orphaned file is a file that exists in the replica data on the target, but does not exist in the protected data on the source. This option specifies if orphaned files should be deleted on the target.

      Orphaned file configuration is a per target configuration. All jobs to the same target will have the same orphaned file configuration.

      If delete orphaned files is enabled, carefully review any replication rules that use wildcard definitions. If you have specified wildcards to be excluded from protection, files matching those wildcards will also be excluded from orphaned file processing and will not be deleted from the target. However, if you have specified wildcards to be included in your protection, those files that fall outside the wildcard inclusion rule will be considered orphaned files and will be deleted from the target.

Snapshots



Snapshots

Disk space allocated for each snapshot

20 percent of the original LVM volume size

□ Enable scheduled snapshots

Take snapshots on this interval: 1 Hours

● Begin immediately

○ Begin at this time: 11/20/2018 9:47:33 AM

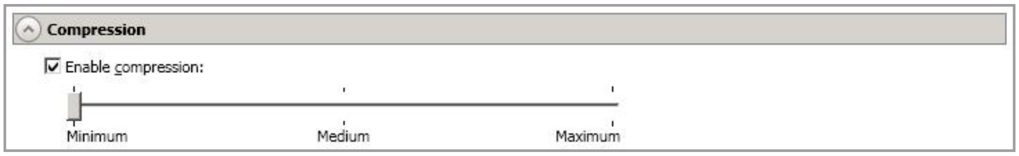
Snapshots are not supported on Btrfs file systems or on servers running Ubuntu 12.04.x. See snapshots in *Full server requirements* on page 106 for complete details on the snapshot requirements.

A snapshot is an image of the source replica data on the target taken at a single point in time. You can failover to a snapshot. However, you cannot access the snapshot to recover specific files or folders.

- **Disk space allocated for each snapshot**—When you take a snapshot (either scheduled or manual), Carbonite Availability needs to know how much disk space to allocate for each one. Specify a percentage of the original LVM volume size to allocate for each snapshot. Once a snapshot is created, you can see the percentage of allocated space that has been consumed by using the lvs command. This is the LVM utility command to list the information for each logical volume on the server. Check the Data% column to see the space consumed. Once the percentage of allocated space reaches 100%, LVM cannot use the snapshot. Before you get to 100%, you can manually expand the snapshot as long as there is enough space in the volume group. See your Linux documentation for details on expanding the snapshot.
- **Enabled scheduled snapshots**—Select this option if you want Carbonite Availability to take snapshots automatically at set intervals.
  - **Take snapshots on this interval**—Specify the interval (in days, hours, or minutes) for taking snapshots.
  - **Begin immediately**—Select this option if you want to start taking snapshots immediately after the protection job is established.
  - **Begin at this time**—Select this option if you want to start taking snapshots at a later date and time. Specify the date and time parameters to indicate when you want to start.

See *Managing snapshots* on page 59 for details on taking manual snapshots and deleting snapshots.

Compression



Compression

☑ Enable compression:

Minimum Medium Maximum

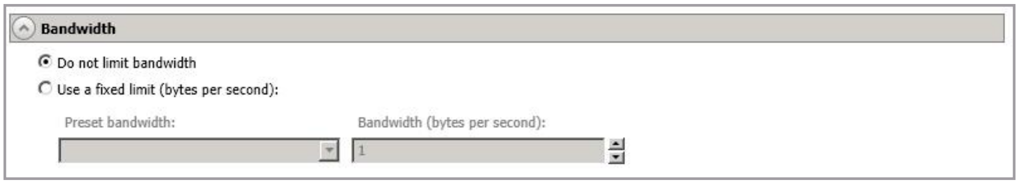
To help reduce the amount of bandwidth needed to transmit Carbonite Availability data, compression allows you to compress data prior to transmitting it across the network. In a WAN environment this provides optimal use of your network resources. If compression is enabled, the data is compressed before it is transmitted from the source. When the target receives the compressed data, it decompresses it and then writes it to disk. You can set the level from **Minimum** to **Maximum** to suit your needs.

Keep in mind that the process of compressing data impacts processor usage on the source. If you notice an impact on performance while compression is enabled in your environment, either adjust to a lower level of compression, or leave compression disabled. Use the following guidelines to determine whether you should enable compression.

- If data is being queued on the source at any time, consider enabling compression.
- If the server CPU utilization is averaging over 85%, be cautious about enabling compression.
- The higher the level of compression, the higher the CPU utilization will be.
- Do not enable compression if most of the data is inherently compressed. Many image (.jpg, .gif) and media (.wmv, .mp3, .mpg) files, for example, are already compressed. Some images files, such as .bmp and .tif, are decompressed, so enabling compression would be beneficial for those types.
- Compression may improve performance even in high-bandwidth environments.
- Do not enable compression in conjunction with a WAN Accelerator. Use one or the other to compress Carbonite Availability data.

All jobs from a single source connected to the same IP address on a target will share the same compression configuration.

Bandwidth



Bandwidth

● Do not limit bandwidth

○ Use a fixed limit (bytes per second):

Preset bandwidth:

Bandwidth (bytes per second): 1

Bandwidth limitations are available to restrict the amount of network bandwidth used for Carbonite Availability data transmissions. When a bandwidth limit is specified, Carbonite Availability never exceeds that allotted amount. The bandwidth not in use by Carbonite Availability is available for all other network traffic.

> All jobs from a single source connected to the same IP address on a target will share the same bandwidth configuration.

- **Do not limit bandwidth**—Carbonite Availability will transmit data using 100% bandwidth availability.
- **Use a fixed limit**—Carbonite Availability will transmit data using a limited, fixed bandwidth. Select a **Preset bandwidth** limit rate from the common bandwidth limit values. The **Bandwidth** field will automatically update to the bytes per second value for your selected bandwidth. This is the maximum amount of data that will be transmitted per second. If desired, modify the bandwidth using a bytes per second value. The minimum limit should be 3500 bytes per second.

8. Click **Next** to continue.
9. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

   Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

   If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

   After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Migrate Reference Guide* for details on the various Carbonite Availability log files.
10. Once your servers have passed validation and you are ready to establish protection, click **Finish**,

and you will automatically be taken to the **Jobs** page.

Jobs in a NAT environment may take longer to start.

Once a job is created, do not change the name of underlying hardware components used in the job. For example, volume and datastore names or network adapter and virtual switch names. Any component used by name in your job must continue to use that name throughout the lifetime of job. If you must change a name, you will need to delete the job and re-create it using the new component name.

Managing and controlling full server jobs

Click **Jobs** from the main Carbonite Replication Console toolbar. The **Jobs** page allows you to view status information about your jobs. You can also control your jobs from this page.

The jobs displayed in the right pane depend on the server group folder selected in the left pane. Every job for each server in your console session is displayed when the **Jobs on All Servers** group is selected. If you have created and populated server groups (see *Managing servers* on page 24), then only the jobs associated with the server or target servers in that server group will be displayed in the right pane.

- *Overview job information displayed in the top right pane* on page 133
- *Detailed job information displayed in the bottom right pane* on page 136
- *Job controls* on page 138

Overview job information displayed in the top right pane

The top pane displays high-level overview information about your jobs. You can sort the data within a column in ascending and descending order. You can also move the columns to the left or right of each other to create your desired column order. The list below shows the columns in their default left to right order.

If you are using server groups, you can filter the jobs displayed in the top right pane by expanding the **Server Groups** heading and selecting a server group.

Column 1 (Blank)

The first blank column indicates the state of the job.

✔ A green circle with a white checkmark indicates the job is in a healthy state. No action is required.

⚠ A yellow triangle with a black exclamation point indicates the job is in a pending or warning state. This icon is also displayed on any server groups that you have created that contain a job in a pending or warning state. Carbonite Availability is working or waiting on a pending process or attempting to resolve the warning state.

✖ A red circle with a white X indicates the job is in an error state. This icon is also displayed on any server groups that you have created that contain a job in an error state. You will need to investigate and resolve the error.

? The job is in an unknown state.

Job

The name of the job

Source Server

The name of the source. This could be the name or IP address of your source.

Target Server

The name of the target. This could be the name or IP address of your target.

Job Type

Each job type has a unique job type name. This job is a Full Server job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the job details. Keep in mind that **Idle** indicates console to server activity is idle, not that your servers are idle.

Mirror Status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.
- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown**—The console cannot determine the status.

Replication Status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Transmit Mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Operating System

The job type operating system

Detailed job information displayed in the bottom right pane

The details displayed in the bottom pane provide additional information for the job highlighted in the top pane. You can expand or collapse the bottom pane by clicking on the **Job Highlights** heading.

Name

The name of the job

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready—**The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the recovery point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the recovery point latency time.

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent (compressed)

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Connected since

The date and time indicating when the current job was started.

Recent activity

Displays the most recent activity for the selected job, along with an icon indicating the success or failure of the last initiated activity. Click the link to see a list of recent activities for the selected job. You can highlight an activity in the list to display additional details about the activity.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Job controls

You can control your job through the toolbar buttons available on the **Jobs** page. If you select multiple jobs, some of the controls will apply only to the first selected job, while others will apply to all of the selected jobs. For example, **View Job Details** will only show details for the first selected job, while **Stop** will stop protection for all of the selected jobs.

If you want to control just one job, you can also right click on that job and access the controls from the pop-up menu.

View Job Details

This button leaves the **Jobs** page and opens the **View Job Details** page.

Edit Job Properties

This button leaves the **Jobs** page and opens the **EditJob Properties** page.

Delete

Stops (if running) and deletes the selected jobs.

Provide Credentials

Changes the login credentials that the job (which is on the target machine) uses to authenticate to the servers in the job. This button opens the Provide Credentials dialog box where you can specify the new account information and which servers you want to update.

View Recent Activity

Displays the recent activity list for the selected job. Highlight an activity in the list to display additional details about the activity.

Start

Starts or resumes the selected jobs.

If you have previously stopped protection, the job will restart mirroring and replication.

If you have previously paused protection, the job will continue mirroring and replication from where it left off, as long as the Carbonite Availability queue was not exhausted during the time the job was paused. If the Carbonite Availability queue was exhausted during the time the job was paused, the job will restart mirroring and replication.

Also if you have previously paused protection, all jobs from the same source to the same IP address on the target will be resumed.

Pause

Pauses the selected jobs. Data will be queued on the source while the job is paused. Failover monitoring will continue while the job is paused.

All jobs from the same source to the same IP address on the target will be paused.

Stop

Stops the selected jobs. The jobs remain available in the console, but there will be no mirroring or replication data transmitted from the source to the target. Mirroring and replication data will not be queued on the source while the job is stopped, requiring a remirror when the job is restarted. The type of remirror will depend on your job settings. Failover monitoring will continue while the job is stopped.

Take Snapshot

Even if you have scheduled the snapshot process, you can run it manually any time. If an automatic or scheduled snapshot is currently in progress, Carbonite Availability will wait until that one is finished before taking the manual snapshot.

Snapshots are not supported on Btrfs file systems or on servers running Ubuntu 12.04.x. See snapshots in *Full server requirements* on page 106 for complete details on the snapshot requirements.

Manage Snapshots

Allows you to manage your snapshots by taking and deleting snapshots for the selected job. See *Managing snapshots* on page 59 for more information.

Failover or Cutover

Starts the failover process. See *Failing over full server jobs* on page 150 for the process and details of failing over a Linux full server job.

Failback

Starts the failback process. Failback does not apply to full server for Linux jobs.

Restore

Starts the restoration process. Restoration does not apply to full server for Linux jobs.

Reverse

Reverses protection. The original source hardware will be reversed to the target identity and the job will start mirroring in the reverse direction with the job name and log file names changing accordingly. After the mirror is complete, the job will continue running in the opposite direction. See *Reversing full server jobs* on page 152 for the process and details of reversing a full server job.

Undo Failover or Cutover

Cancels a test failover by undoing it. Undo failover does not apply to full server for Linux jobs.

View Job Log

Opens the job log. On the right-click menu, this option is called **View Logs**, and you have the option of opening the job log, source server log, or target server log.

Other Job Actions

Opens a small menu of other job actions. These job actions are not available for Linux jobs.

Generate Activity Report

For all Windows jobs except files and folders, you can create a basic failover report. This is the same report created by Get-DtLatestFailoverReport and Get-DtAllFailoverReports from the Carbonite Availability PowerShell module. The report will be located on the target in the \Service\Reports directory where Carbonite Availability is installed. For Linux jobs, you must use PowerShell.

Filter

Select a filter option from the drop-down list to only display certain jobs. You can display **Healthy jobs**, **Jobs with warnings**, or **Jobs with errors**. To clear the filter, select **All jobs**. If you have created and populated server groups, then the filter will only apply to the jobs associated with the server or target servers in that server group. See *Managing servers* on page 24.

Search

Allows you to search the source or target server name for items in the list that match the criteria you have entered.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Viewing full server job details

From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.

Review the following table to understand the detailed information about your job displayed on the **View Job Details** page.

Job name

The name of the job

Job type

Each job type has a unique job type name. This job is a Full Server job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Health

✅ The job is in a healthy state.

⚠ The job is in a warning state.

❌ The job is in an error state.

? The job is in an unknown state.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the job details.

Connection ID

The incremental counter used to number connections. The number is incremented when a connection is created. The counter is reset if there are no existing jobs and the Double-Take service is restarted.

Transmit mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Target route

The IP address on the target used for Carbonite Availability transmissions.

Compression

- **On** / **Level**—Data is compressed at the level specified.
- **Off**—Data is not compressed.

Encryption

- **On**—Data is being encrypted before it is sent from the source to the target.
- **Off**—Data is not being encrypted before it is sent from the source to the target.

Bandwidth limit

If bandwidth limiting has been set, this statistic identifies the limit. The keyword **Unlimited** means there is no bandwidth limit set for the job.

Connected since

The source server date and time indicating when the current job was started. This field is blank, indicating that a TCP/IP socket is not present, when the job is waiting on transmit options or if the transmission has been stopped. This field will maintain the date and time, indicating that a TCP/IP socket is present, when transmission has been paused.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Mirror status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.

- **Idle**—Data is not being mirrored.
- **Paused—**Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans—**Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown**—The console cannot determine the status.

Mirror percent complete

The percentage of the mirror that has been completed

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory—**Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent compressed

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the recovery point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the recovery point latency time.

Mirror start time

The UTC time when mirroring started

Mirror end time

The UTC time when mirroring ended

Total time for last mirror

The length of time it took to complete the last mirror process

Target Server Image

When a full server job is created with reverse protection enabled, an image of the target's system state is stored on the source server. This image allows you to reverse your source and target after a failover. To improve performance, the target's system state is not continuously replicated to the source. You should manually update the image of the target's system state by clicking **Update** if there is a change on the target. For example, if the credentials on the target server are updated, you should update the target server image that is on the source. This reverse protection mirror may cause a performance impact on your source server. This impact is only temporary, and system performance will return to normal when the reverse protection mirror is complete.

If you have reverse enabled, are updating your target image, and the Double-Take service on the target restarts (either manually or automatically, for example the target server restarts), you should restart your target image update after the Double-Take service is back online. This will correct any incorrect status displayed in the console and ensure the target image is complete.

Validating a full server job

Over time, you may want to confirm that any changes in your network or environment have not impacted your Carbonite Availability job. Use these instructions to validate an existing job.

1. From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.
2. In the **Tasks** area on the right on the **View Job Details** page, click **Validate job properties**.
3. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

   Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

   Validation checks for an existing job are logged to the job log on the target server.
4. Once your servers have passed validation, click **Close**.

Editing a full server job

Use these instructions to edit a full server job.

1. From the **Jobs** page, highlight the job and click **Edit Job Properties** in the toolbar. (You will not be able to edit a job if you have removed the source of that job from your Carbonite Replication Console session or if you only have Carbonite Availability monitor security access.)
2. You will see the same options for your full server job as when you created the job, but you will not be able to edit all of them. If desired, edit those options that are configurable for an existing job. See *Creating a full server job* on page 114 for details on each job option.

   Changing some options may require Carbonite Availability to automatically disconnect, reconnect, and remirror the job.

3. If you want to modify the workload items or replication rules for the job, click **Edit workload or replication rules**. Modify the **Workload item** you are protecting, if desired. Additionally, you can modify the specific **Replication Rules** for your job.

   Volumes and folders with a green highlight are included completely. Volumes and folders highlighted in light yellow are included partially, with individual files or folders included. If there is no highlight, no part of the volume or folder is included. To modify the items selected, highlight a volume, folder, or file and click **Add Rule**. Specify if you want to **Include** or **Exclude** the item. Also, specify if you want the rule to be recursive, which indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select **Recursive**, the rule will not be applied to subdirectories.

   You can also enter wildcard rules, however you should do so carefully. Rules are applied to files that are closest in the directory tree to them. If you have rules that include multiple folders, an exclusion rule with a wild card will need to be added for each folder that it needs applied to. For example, if you want to exclude all .log files from /home and your rules include /home, /home/folder1, and /home/folder2, you would need to add the exclusion rule for the root and each subfolder rule. So you will need to add exclude rules for /home/*.log , /home/folder1/*.log, and /home/folder2/*.log.

   If you need to remove a rule, highlight it in the list at the bottom and click **Remove Rule**. Be careful when removing rules. Carbonite Availability may create multiple rules when you are adding directories. For example, if you add /home/admin to be included in protection, then /home will be excluded. If you remove the /home exclusion rule, then the /home/admin rule will be removed also.

   Click **OK** to return to the **Edit Job Properties** page.

   If you remove data from your workload and that data has already been sent to the target, you will need to manually remove that data from the target. Because the data you removed is no longer included in the replication rules, Carbonite Availability orphan file detection cannot remove the data for you. Therefore, you have to remove it manually.

4. Click **Next** to continue.
5. Carbonite Availability validates that your source and target are compatible. The **Summary** page

displays your options and validation items.

Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

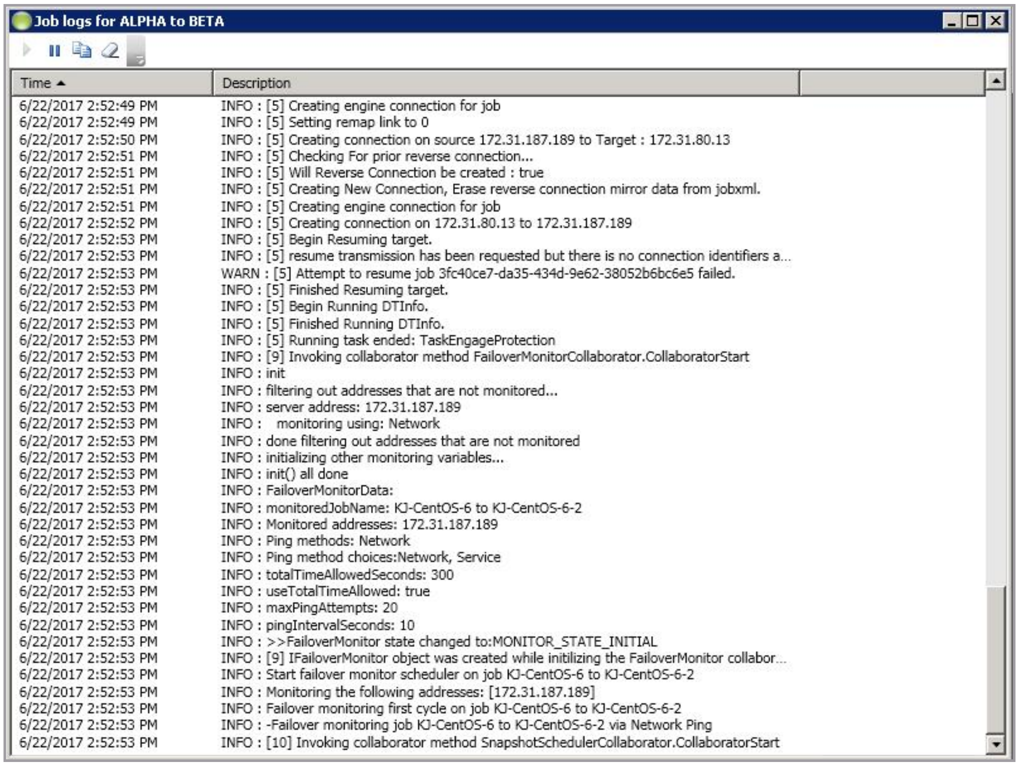
After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Migrate Reference Guide* for details on the various Carbonite Availability log files.

6. Once your servers have passed validation and you are ready to update your job, click **Finish**.

Viewing a full server job log

You can view a job log file through the Carbonite Replication Console by selecting **View Job Log** from the toolbar on the **Jobs** page. Separate logging windows allow you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple jobs. When the Carbonite Replication Console is closed, all logging windows will automatically close.

Because the job log window communicates with the target server, if the console loses communication with the target server after the job log window has already been opened, the job log window will display an error.



```
Time ▲                  Description
6/22/2017 2:52:49 PM    INFO : [5] Creating engine connection for job
6/22/2017 2:52:49 PM    INFO : [5] Setting remap link to 0
6/22/2017 2:52:50 PM    INFO : [5] Creating connection on source 172.31.187.189 to Target : 172.31.80.13
6/22/2017 2:52:51 PM    INFO : [5] Checking For prior reverse connection...
6/22/2017 2:52:51 PM    INFO : [5] Will Reverse Connection be created : true
6/22/2017 2:52:51 PM    INFO : [5] Creating New Connection, Erase reverse connection mirror data from jobxml.
6/22/2017 2:52:51 PM    INFO : [5] Creating engine connection for job
6/22/2017 2:52:52 PM    INFO : [5] Creating connection on 172.31.80.13 to 172.31.187.189
6/22/2017 2:52:53 PM    INFO : [5] Begin Resuming target.
6/22/2017 2:52:53 PM    INFO : [5] resume transmission has been requested but there is no connection identifiers a...
6/22/2017 2:52:53 PM    WARN : [5] Attempt to resume job 3fc40ce7-da35-434d-9e62-38052b6bc6e5 failed.
6/22/2017 2:52:53 PM    INFO : [5] Finished Resuming target.
6/22/2017 2:52:53 PM    INFO : [5] Begin Running DTInfo.
6/22/2017 2:52:53 PM    INFO : [5] Finished Running DTInfo.
6/22/2017 2:52:53 PM    INFO : [5] Running task ended: TaskEngageProtection
6/22/2017 2:52:53 PM    INFO : [9] Invoking collaborator method FailoverMonitorCollaborator.CollaboratorStart
6/22/2017 2:52:53 PM    INFO : init
6/22/2017 2:52:53 PM    INFO : filtering out addresses that are not monitored...
6/22/2017 2:52:53 PM    INFO : server address: 172.31.187.189
6/22/2017 2:52:53 PM    INFO :   monitoring using: Network
6/22/2017 2:52:53 PM    INFO : done filtering out addresses that are not monitored
6/22/2017 2:52:53 PM    INFO : initializing other monitoring variables...
6/22/2017 2:52:53 PM    INFO : init() all done
6/22/2017 2:52:53 PM    INFO : FailoverMonitorData:
6/22/2017 2:52:53 PM    INFO : monitoredJobName: KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM    INFO : Monitored addresses: 172.31.187.189
6/22/2017 2:52:53 PM    INFO : Ping methods: Network
6/22/2017 2:52:53 PM    INFO : Ping method choices:Network, Service
6/22/2017 2:52:53 PM    INFO : totalTimeAllowedSeconds: 300
6/22/2017 2:52:53 PM    INFO : useTotalTimeAllowed: true
6/22/2017 2:52:53 PM    INFO : maxPingAttempts: 20
6/22/2017 2:52:53 PM    INFO : pingIntervalSeconds: 10
6/22/2017 2:52:53 PM    INFO : >>FailoverMonitor state changed to:MONITOR_STATE_INITIAL
6/22/2017 2:52:53 PM    INFO : [9] IFailoverMonitor object was created while initilizing the FailoverMonitor collabor...
6/22/2017 2:52:53 PM    INFO : Start failover monitor scheduler on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM    INFO : Monitoring the following addresses: [172.31.187.189]
6/22/2017 2:52:53 PM    INFO : Failover monitoring first cycle on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM    INFO : -Failover monitoring job KJ-CentOS-6 to KJ-CentOS-6-2 via Network Ping
6/22/2017 2:52:53 PM    INFO : [10] Invoking collaborator method SnapshotSchedulerCollaborator.CollaboratorStart
```

The following table identifies the controls and the table columns in the **Job logs** window.

Start ▶

This button starts the addition and scrolling of new messages in the window.

Pause ⏸

This button pauses the addition and scrolling of new messages in the window. This is only for the **Job logs** window. The messages are still logged to their respective files on the server.

Copy

This button copies the messages selected in the **Job logs** window to the Windows clipboard.

Clear

This button clears the **Job logs** window. The messages are not cleared from the respective files on the server. If you want to view all of the messages again, close and reopen the **Job logs** window.

Time

This column in the table indicates the date and time when the message was logged.

Description

This column in the table displays the actual message that was logged.

Failing over full server jobs

When a failover condition has been met, failover will be triggered automatically if you disabled the wait for user option during your failover configuration. If the wait for user before failover option is enabled, you will be notified in the console when a failover condition has been met. At that time, you will need to trigger it manually from the console when you are ready.

If you have paused your target, failover will not start if configured for automatic failover, and it cannot be initiated if configured for manual intervention. You must resume the target before failover will automatically start or before you can manually start it.

Resolve any maintenance updates on the source that may require the server to be rebooted before failover or failback. Also, do not failover or failback if the target is waiting on a reboot after applying maintenance. If failover occurs before the required reboot, the target may not operate properly or it may not boot.

1. On the **Jobs** page, highlight the job that you want to failover and click **Failover or Cutover** in the toolbar.
2. Select the type of failover to perform.
    - **Failover to live data**—Select this option to initiate a full, live failover using the current data on the target. The source is automatically shut down if it is still running. Then the target will stand in for the source by rebooting and applying the source identity, including its system state, on the target. After the reboot, the target becomes the source, and the target no longer exists.
    - **Perform test failover**—Select this option to perform a test failover.
        - Your source must have / or /opt/dbtk/var/lib on LVM in order to use the test failover feature.
        - All data volumes must be under LVM for test failover.
        - Test failover is not supported for Btrfs file systems.
        - Your test failover server must have the same volume configuration (BIOS or UEFI) as your target server.
        - The source, target, and protection job will remain online and uninterrupted during the test.
        - During the test, any scheduled snapshots for the protection job will be deferred until after the test. Manual snapshots will be disabled until after the test.
        - The test will be performed using the test failover settings configured during job creation.
        - The test failover will take a snapshot of the current data on the target and mirror the data from the snapshot to the test failover machine using the same mirroring options as the protection job.
        - Once the mirror is complete, the test failover machine is rebooted automatically to finalize the test failover process.
        - The test failover machine will maintain its own networking which keeps it isolated from the rest of the network in order to avoid network conflicts and redirecting clients.

    Applications or functionality that relies on the source networking may not be fully testable with the test machine networking.
    - When you are finished with your test, undo it.
    - When you undo a test failover, the snapshot will be deleted.
    - At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)
  - **Failover to a snapshot**—Select this option to initiate a full, live failover without using the current data on the target. Instead, select a snapshot and the data on the target will be reverted to that snapshot. This option will not be available if there are no snapshots on the target. To help you understand what snapshots are available, the **Type** indicates the kind of snapshot.
    - **Scheduled**—This snapshot was taken as part of a periodic snapshot.
    - **Deferred**—This snapshot was taken as part of a periodic snapshot, although it did not occur at the specified interval because the job between the source and target was not in a good state.
    - **Manual**—This snapshot was taken manually by a user.

  Snapshots are not supported on Btrfs file systems or on servers running Ubuntu 12.04.x. See snapshots in *Full server requirements* on page 106 for complete details on the snapshot requirements.

3. Select how you want to handle the data in the target queue.
  - **Apply data in target queues before failover or cutover**—All of the data in the target queue will be applied before failover begins. The advantage to this option is that all of the data that the target has received will be applied before failover begins. The disadvantage to this option is depending on the amount of data in queue, the amount of time to apply all of the data could be lengthy.
  - **Discard data in the target queues and failover or cutover immediately**—All of the data in the target queue will be discarded and failover will begin immediately. The advantage to this option is that failover will occur immediately. The disadvantage is that any data in the target queue will be lost.
4. When you are ready to begin failover, click **Failover**.
5. If you performed a test failover, you can undo it by selecting **Undo Failover or Cutover** in the toolbar. If configured, the snapshots used for the test failover will be deleted.

If you need to update DNS after failover, there is a sample DNS update script located in /etc/DT/sysprep.d. You may need to modify the script for your environment. If you need basic assistance with script modifications, contact technical support. Assistance with advanced scripting will be referred to Professional Services.

Reversing full server jobs

After a full server failover, the source is running on your original target hardware and your target no longer exists. That means the source and target hardware now share the same identity, which is the source identity.

If you did not enable reverse protection or if you have to rebuild your source, you will have to reverse your protection manually.

1. Fix the issue that caused your original source server to fail.
2. Connect the original source server to the network.
3. Make sure the production NIC on your original source is online. If the NIC is disabled or unplugged, you will not be able to reverse. Make sure you continue to access the servers through the reserved IP addresses, but you can disregard any IP address conflicts for the primary NIC. Since the new source (running on the original target hardware) already has the source's address assigned to it, the source reserved IP address (set during the job creation workflow) will be used to identify the source. The machine names for both servers will be the same at this point. The reserved IP addresses which were selected during the job creation will be shown in parenthesis to identify the machines.
4. On the **Jobs** page, highlight the job that you want to reverse. If the job is not listed, you may need to add your servers to your console again. Use the reserved IP addresses and local credentials.
5. Highlight the job you want to reverse and click **Reverse** in the toolbar. During the reverse process, you will see various states for the job. The **Reversing** state will be displayed when the target identity is being established on the original source hardware. When the reverse process is complete, the target (on the original source hardware) will reboot. At this point, your source is still running on your original target hardware with the source name, but the original source hardware now has the target identity. After reboot, the job will start synchronizing. During the synchronizing process, protection is being established from the source (on the original target hardware) to the target (on the original source hardware). The reverse protection is also established in the opposite direction.
6. To go back to your original hardware, highlight the job and click **Failover or Cutover**. The source identity will now be applied to the target (on the original source hardware), and the target identity will again be gone. Both servers will have the source identity.
7. To bring back the target identity, highlight the job and click **Reverse**. The same process as above will be repeated, but on the opposite servers. When the reverse is complete, you will be back to your original identities on the original hardware.

Chapter 6 Full server to ESX protection

Create a full server to ESX job when you want to protect an entire physical server or virtual machine to an ESX target. There is no reverse protection for this job. You will have to use another full server job type to get back to your original hardware after failover.

For full server to ESX protection, you will need to complete the following steps, in order.

1. Review the *Full server to ESX requirements* on page 154 to make sure your environment meets the requirements.
2. Deploy your Carbonite Availability virtual recovery appliance. See the *Carbonite Availability and Carbonite Migrate Installation, Licensing, and Activation* document for details.
3. Install the Carbonite Replication Console on a Windows machine. See the *Carbonite Availability and Carbonite Migrate Installation, Licensing, and Activation* document for details.
4. Add your virtual recovery appliance to the Carbonite Replication Console. See *Adding servers* on page 34.
5. Install Carbonite Availability on your Linux source servers. See the *Carbonite Availability and Carbonite Migrate Installation, Licensing, and Activation* document for details.
6. Add your source servers to your Carbonite Replication Console. See *Adding servers* on page 34.
7. Create your full server to ESX appliance job. See *Creating a full server to ESX job* on page 162.

Once your job is created and running, see the following sections to manage your job.

- *Managing and controlling full server to ESX jobs* on page 189—You can view status information about your job and learn how to control the job.
- *Failing over full server to ESX jobs* on page 206—Use this section when a failover condition has been met or whenever you want to failover.
- *Reversing protection after failover for full server to ESX jobs* on page 209—Use this section if you need to get your data back to the original hardware.

Full server to ESX requirements

Use these requirements for full server to ESX protection.

- **Source server**—The source server can be a physical or virtual server running any of the following operating systems.
  - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
    - **Version**—5.9 through 5.11
    - **Kernel type for x86 (32-bit) architectures**—Default, SMP, Xen, PAE
    - **Kernel type for x86-64 (64-bit) architectures**—Default, SMP, Xen
    - **File system**—Ext3, Ext4, XFS
    - **Notes**—Oracle Enterprise Linux support is for the mainline kernel only, not the Unbreakable kernel.
  - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
    - **Version**—6.8 through 6.10
    - **Kernel type for x86 (32-bit) architectures**—Default
    - **Kernel type for x86-64 (64-bit) architectures**—Default
    - **File system**—Ext3, Ext4, XFS (64-bit only)
  - **Operating system**—Red Hat Enterprise Linux, Oracle Enterprise Linux, and CentOS
    - **Version**—7.3 through 7.5
    - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
    - **Kernel type for x86-64 (64-bit) architectures**—Default
    - **File system**—Ext3, Ext4, XFS
  - **Operating system**—SUSE Linux Enterprise
    - **Version**—11.2 through 11.4
    - **Kernel type for x86 (32-bit) architectures**—Default, Xen, XenPAE, VMI
    - **Kernel type for x86-64 (64-bit) architectures**—Default, Xen
    - **File system**—Ext3, XFS
  - **Operating system**—SUSE Linux Enterprise
    - **Version**—12.1 through 12.3
    - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
    - **Kernel type for x86-64 (64-bit) architectures**—Default
    - **File system**—Ext3, Ext4, XFS, Btrfs
    - **Notes**—If you are planning to convert an existing file system to Btrfs, you must delete any existing Carbonite Availability jobs and re-create them after converting to Btrfs.
  - **Operating system**—Ubuntu
    - **Version**—12.04.3, 12.04.4, and 12.04.5
    - **Kernel type for x86 (32-bit) architectures**—Generic

    - **Kernel type for x86-64 (64-bit) architectures**—Generic
    - **File system**—Ext2, Ext3, Ext4, XFS
  - **Operating system**—Ubuntu
    - **Version**—14.04.3, 14.04.4, and 14.04.5
    - **Kernel type for x86 (32-bit) architectures**—Generic
    - **Kernel type for x86-64 (64-bit) architectures**—Generic
    - **File system**—Ext2, Ext3, Ext4, XFS
  - **Operating system**—Ubuntu
    - **Version**—16.04.2, 16.04.3, and 16.04.4
    - **Kernel type for x86 (32-bit) architectures**—Generic
    - **Kernel type for x86-64 (64-bit) architectures**—Generic
    - **File system**—Ext2, Ext3, Ext4, XFS
  - **Operating system**—Ubuntu
    - **Version**—18.04.0
    - **Kernel type for x86 (32-bit) architectures**—No 32-bit architectures are supported
    - **Kernel type for x86-64 (64-bit) architectures**—Generic
    - **File system**—Ext2, Ext3, Ext4, XFS

For all operating systems except Ubuntu, the kernel version must match the expected kernel for the specified release version. For example, if /etc/redhat-release declares the system to be a Redhat 7.5 system, the kernel that is installed must match that.

Carbonite Availability does not support stacking filesystems, like eCryptFS.

If Carbonite Availability does not have the driver binary files for the kernel you are using, they can be compiled automatically, but you need the build-essential package for them to be installed. Run apt-get install build-essential to install the build tools and then restart the DT service. This will build the driver from the source and load it.

- **Packages and services**—Each Linux server must have the following packages and services installed before you can install and use Carbonite Availability. See your operating system documentation for details on these packages and utilities.
  - sshd (or the package that installs sshd)
  - lsb
  - parted
  - dmidecode
  - scp
  - which
- **vCenter**—vCenter is not required, but if you are using it, then you must use version 5.5 or later. If you upgrade your version of vCenter after it has been entered into the Carbonite Replication Console, you must remove and re-add the vCenter in order for the console to recognize the upgraded version.
- **vMotion**—Host vMotion is only supported if you are using vCenter. Storage vMotion is not

supported.

- **Target host server**—The target host server must be an ESX server. It can be any of the following ESX operating systems.
  - ESXi 5.5
  - ESXi 6.0
  - ESXi 6.5
  - ESXi 6.7

  The free versions of ESX restrict functionality that Carbonite Availability requires. Therefore, you must use one of the paid editions of ESX.

- **Virtual recovery appliance**—The target ESX host must have an existing virtual machine, known as a virtual recovery appliance. You must have this appliance before you can begin protection. When you begin protection, the virtual recovery appliance will mount disks, format disks, and so on. When failover occurs, a new virtual machine is powered on using the replicated disks from the appliance. Once the new virtual machine is online, it will have the identity, data, and system state of the source. Since the appliance maintains its own identity, it can be reused for additional failovers.

  You have the choice of using an OVF (Open Virtualization Format) virtual machine included with Carbonite Availability for your appliance, or creating your own appliance that meets the requirements below. In either case, keep in mind the following caveats for the appliance.

  - The virtual recovery appliance must be a standalone virtual machine.
  - It should not reside in any multiple virtual machine vApp.
  - The OVF appliance is pre-configured for optimal performance. You do not need to modify the memory, CPU, or other configurations.
  - You should not install or run anything else on the appliance.
  - The firewall is disabled on the OVF appliance and should remain disabled.
  - A single virtual recovery appliance can protect a maximum of 59 volume groups and raw block devices (combined) from any number of sources.

  If you are creating your own appliance, it must meet the following requirements.

  - **Operating system**—The virtual machine must be running a 64-bit version of one of the following operating systems.
    - Ubuntu 16.04.4
    - Red Hat Enterprise Linux or CentOS version 7.3 through 7.5
    - SUSE Linux Enterprise version 12.1 through 12.3

    A SLES appliance can only protect source servers running a Carbonite Availability supported SLES version. You cannot protect other Linux operating systems to a SLES appliance.

    You cannot protect Btrfs to a Red Hat or CentOS appliance.

  - **Memory**—The virtual machine must have at least 4 GB of virtualized physical RAM.

    - **CPUs**—The virtual machine must have at least two CPUs (two virtual sockets, not two virtual cores).
    - **Disk space**—The virtual machine must have at least 16 GB of disk space available.
    - **Networking**—The virtual machine must have a valid, working network configuration, including DNS.
    - **Function**—The virtual machine must be dedicated to Carbonite Availability processing only. Do not use the virtual machine for any other activity (web server, database server, and so on).
    - **Volume group name**—If your virtual machine is running Red Hat or CentOS and is using an LVM setup, you must make sure the volume group on the virtual machine is using a unique name. If the same volume group name is used as any volume group name from a protected source, failover will fail because of a name conflict. Refer to your Red Hat documentation for details on renaming a volume group.
    - **Packages**—You will need specific packages installed on your appliance depending on the operating system of your source servers.
        - **Ext**—If the source server you will be protecting has the ext file system, you must have the e2fsprogs package on your appliance.
        - **Xfs**—If the source server you will be protecting has the xfs file system, you must have the xfsprogs package on your appliance.
        - **LVM**—If the source server you will be protecting has an LVM setup, you must have the lvm2 package on your appliance.
        - **Btrfs**—If the source server you will be protecting has the Btrfs file system and you are using an Ubuntu appliance, the appliance must have the btrfs-tools package. If the source server you are protecting is SLES 12.x with Btrfs and you are using a SLES appliance, the btrfsprogs package should already be on the SLES appliance by default. You cannot protect Btrfs to a Red Hat or CentOS appliance.
- **Permissions**—If you want to limit the permissions required for the account that you will be using for your full server to ESX job, your account must have at a minimum the permissions listed below. These permissions can be set at the vCenter, Datacenter, or host level.
    - **Datastore**—Allocate Space, Browse Datastore, Low level file operations, and Remove File
    - **Host**, **Local Operations**—Create Virtual Machine, Delete Virtual Machine, and Reconfigure virtual machine
    - **Network**—Assign Network
    - **Resource**—Assign virtual machine to resource pool
    - **Scheduled Task**—Create Tasks, Modify Task, Remove Task, and Run Task
    - **Tasks**—Create task and Update task
    - **Virtual Machine**, **Configuration**—Add existing disk, Add new disk, Add or remove device, Change resource, Modify device settings, and Remove disk
    - **Virtual Machine**, **Interaction**—Device connection, Power off, and Power on
    - **Virtual Machine**, **Inventory**—Create new, Register, Remove, and Unregister

  Make sure if you also define permissions at the VMs and Templates level in vCenter that you have not denied any of the required permissions listed above.
- **System memory**—The minimum system memory on each server is 1 GB.

- **Disk space for program files**—This is the amount of disk space needed for the Carbonite Availability program files. This is approximately 400 MB on a Linux source server. The appliance needs approximately 620 MB.

  Make sure you have additional disk space for Carbonite Availability queuing, logging, and so on.

- **Server name**—Carbonite Availability includes Unicode file system support, but your server name must still be in ASCII format. Additionally, all Carbonite Availability servers and appliances must have a unique server name.
- **Target drivers**—Install on the source any drivers that are required on the target after failover. For example, you need to install on the source any NIC drivers that will be required on the target after failover.
- **Protocols and networking**—Your servers must meet the following protocol and networking requirements.
  - Your servers must have TCP/IP with static IP addressing.
  - IPv4 is the only supported version.
  - If you are using Carbonite Availability over a WAN and do not have DNS name resolution, you will need to add the host names to the local hosts file on each server running Carbonite Availability.
  - Ubuntu Netplan is not a supported configuration. You must be running NetworkManager natively without Netplan. If you protect and failover a Linux server with Netplan, you will have to configure networking manually after failover.
- **NAT support**—Carbonite Availability supports NAT environments with the following caveats.
  - Only IPv4 is supported.
  - Only standalone servers are supported.
  - Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console.
  - DNS failover and updates will depend on your configuration
    - Only the source or target can be behind a router, not both.
    - The DNS server must be routable from the target
- **Name resolution**—Your servers must have name resolution or DNS. The Carbonite Replication Console must be able to resolve the virtual recovery appliance, and the virtual recovery appliance must be able to resolve all source servers. For details on name resolution options, see your Linux documentation or online Linux resources.
- **Ports**—Port 1501 is used for localhost communication between the engine and management service and should be opened inbound and outbound for both TCP and UDP in iptables. Ports 1500, 1505, 1506, 6325, and 6326 are used for component communication and must be opened inbound and outbound for both TCP and UDP on any firewall that might be in use.
- **Security**—Carbonite Availability security is granted through membership in user groups. The

groups can be local or LDAP (Lightweight Directory Access Protocol). A user must provide a valid local account that is a member of the Carbonite Availability security groups.

- **SELinux policy**—SELinux must be disabled on the target appliance. It can be enabled on your source.
- **UEFI, trusted boot, secure boot**—The source boot mode cannot be UEFI (Unified Extensible Firmware Interface), trusted boot (tboot), secure boot, or other volume blocking mechanisms.
- **Docker**—Your source cannot be a Docker host.
- **Mount option**—The mount option noexec is not supported on the /tmp filesystem.
- **Snapshots**—You can take and failover to snapshots using a full server to ESX job. Because Carbonite Availability uses VMware for snapshot capabilities, you must be aware of the requirements and limitations imposed by VMware. See VMware Knowledge Base article 1025279 at kb.vmware.com/kb/1025279 for details on best practices for VMware snapshots. Additionally, you cannot reuse a virtual disk if it has snapshots associated with it. You must delete all snapshots before you can reuse a virtual disk.
- **Test failover**—Test failover allows you to keep your job intact and use a third machine to test the failover process. To complete the test functionality, Carbonite Availability use LVM snapshots. Keep in mind the following for using test failover.
    - All data volumes must be under LVM for test failover.
    - In order to use test failover, you must make sure your target has sufficient free disk space available. The free space on each volume group on the target must be larger than 50% of the total size of all logical volumes in that volume group on the source. Meeting that amount of free space may depend on the **Disk Configuration Strategy** you selected under **Replica Virtual Disk Volumes**.
        - **Create disks matching source**—With the match source option, you must have sufficient free space on the source before the job is created because the target disks will be matching the source. You may need to increase free disk space on the source (perhaps add a partition which has been created on the raw disk on the source and then extend the volume group), in order to allow for sufficient free space to be matched on the target.
        - **Create disks per volume**—With the per volume option, select a volume group size on the target that has enough free space to accommodate the 50% free space requirement.
    - Test failover is not supported for Btrfs file systems.
    - The source, target, and protection job will remain online and uninterrupted during the test.
    - The test will be performed using the test failover settings configured during job creation.
    - The test will use the current data on the target.
    - Scheduled snapshots will be deferred during the test and taken automatically after the test is undone.
    - The test failover will take a snapshot of the current data on the target and create a new set of virtual disks. The data from the snapshot will be mirrored to the new set of disks using the same mirroring options as the protection job.
    - Once the mirror is complete, the replica virtual machine or alternate replica virtual machine, depending on your selected configuration, is automatically brought online using the new set of disks.

- The replica virtual machine or alternate replica virtual machine, depending on your selected configuration, will use the network settings specified in the test failover settings of the protection job.
- When you are finished with your test, undo it.
- When you undo a test failover, the new set of disks will be maintained or deleted as specified in the test failover settings of the protection job.
- At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)

- **Supported configurations**—The following table identifies the supported configurations for a full server to ESX job.

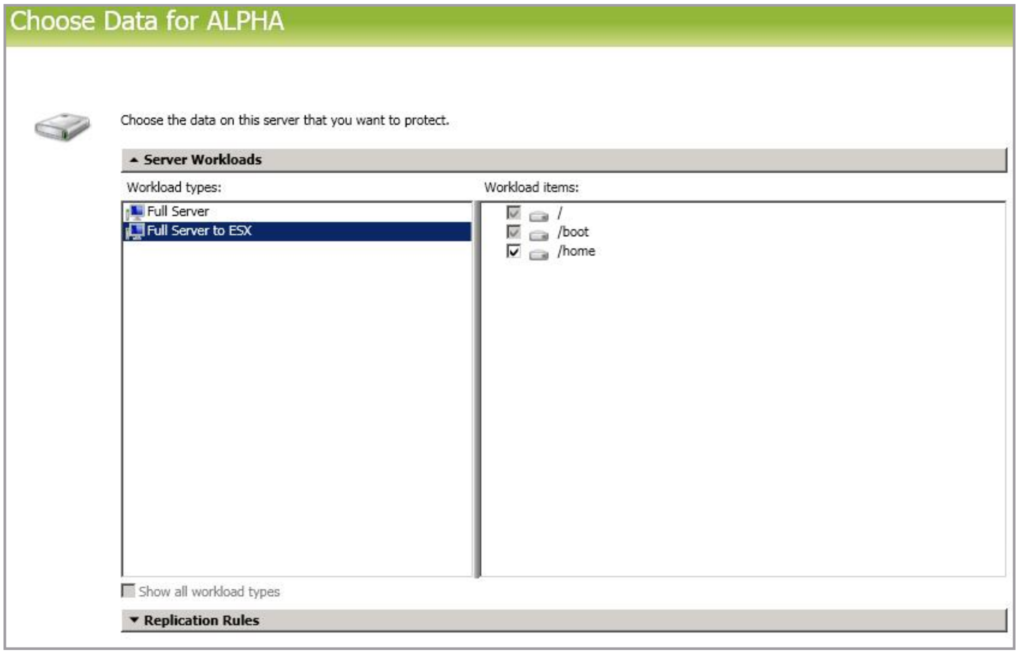
Server to Host Configuration	Description	Supported	Not Supported
One to one active/standby	You can protect a single source to a single target host.	X	
One to one active/active	This configuration (where both the source and target use the same job type to actively replicate to each other) is not supported and not applicable because the target is a hypervisor host.		X
Many to one	You can protect many source servers to one target host. Replication occurs from each source to the one target host. This will consolidate your source servers to a single host.	X	
One to many	You cannot protect a single source to multiple target hosts.		X
Chained	This configuration (where the source replicates to the target and then the target uses the same job type to replicate the source to a final target) is not supported and not applicable because the middle target is a hypervisor host.		X
Single server	You cannot protect a single source to itself.		X
Standalone to standalone	Your source and target host can be in a standalone to standalone configuration.	X	
Standalone to cluster	Your source and target host cannot be in a standalone to cluster configuration.		X
Cluster to standalone	Your source and target host cannot be in a cluster to standalone configuration.		X

Server to Host Configuration	Description	Supported	Not Supported
Cluster to cluster	Your source and target host cannot be in a cluster to cluster configuration.		X

Creating a full server to ESX job

Use these instructions to create a full server to ESX job.

1. From the **Servers** page, right-click the server you want to protect and select **Protect**. You can also highlight a server, click **Create a New Job** in the toolbar, then select **Protect**.
2. Choose the type of workload that you want to protect. Under **Server Workloads**, in the **Workload types** pane, select **Full Server to ESX**. In the **Workload items** pane, select the volumes on the source that you want to protect.



Unsupported file systems will be displayed but will not be accessible.

3. By default, Carbonite Availability selects the system and boot volumes for protection. You will be unable to deselect these volumes. Select any other volumes on the source that you want to protect.

The swap partition is excluded by default and you cannot select it, however, a swap partition will be created on the replica.

If desired, click the **Replication Rules** heading and expand the volumes under **Folders**. You will see that Carbonite Availability automatically excludes particular files that cannot be used during the protection. If desired, you can exclude other files that you do not want to protect, but be careful when excluding data. Excluded volumes, folders, and/or files may compromise the integrity of your installed applications.

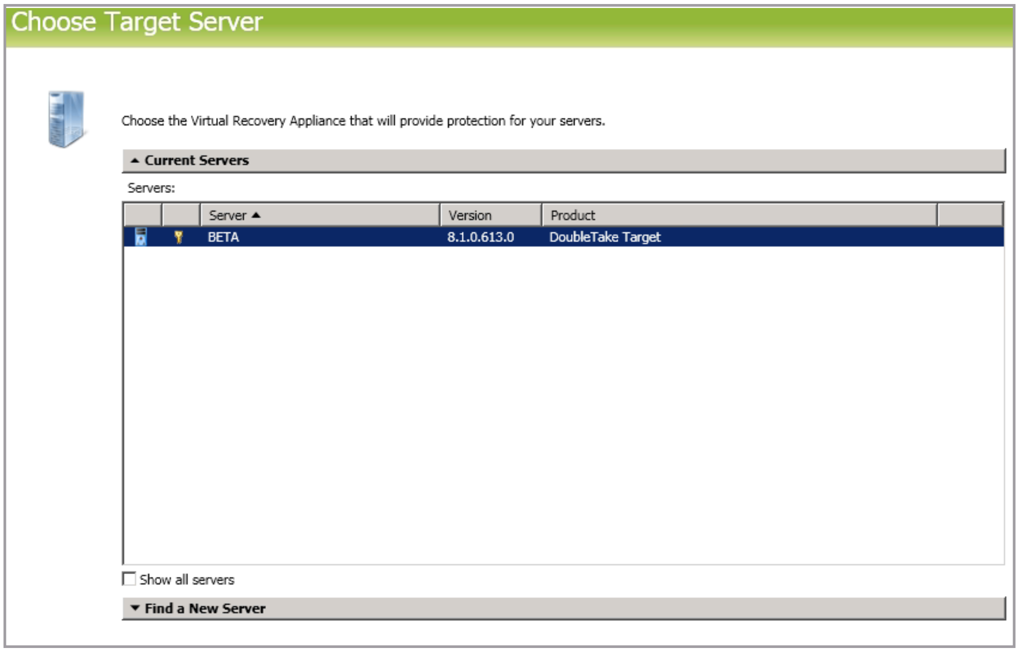
Volumes and folders with a green highlight are included completely. Volumes and folders highlighted in light yellow are included partially, with individual files or folders included. If there is no highlight, no part of the volume or folder is included. To modify the items selected, highlight a volume, folder, or file and click **Add Rule**. Specify if you want to **Include** or **Exclude** the item. Also, specify if you want the rule to be recursive, which indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select **Recursive**, the rule will not be applied to subdirectories.

You can also enter wildcard rules, however you should do so carefully. Rules are applied to files that are closest in the directory tree to them. If you have rules that include multiple folders, an exclusion rule with a wild card will need to be added for each folder that it needs applied to. For example, if you want to exclude all .log files from /home and your rules include /home, /home/folder1, and /home/folder2, you would need to add the exclusion rule for the root and each subfolder rule. So you will need to add exclude rules for /home/*.log , /home/folder1/*.log, and /home/folder2/*.log.

If you need to remove a rule, highlight it in the list at the bottom and click **Remove Rule**. Be careful when removing rules. Carbonite Availability may create multiple rules when you are adding directories. For example, if you add /home/admin to be included in protection, then /home will be excluded. If you remove the /home exclusion rule, then the /home/admin rule will be removed also.

If you return to this page using the **Back** button in the job creation workflow, your **Workload Types** selection will be rebuilt, potentially overwriting any manual replication rules that you specified. If you do return to this page, confirm your **Workload Types** and **Replication Rules** are set to your desired settings before proceeding forward again.

4. Click **Next** to continue.
5. Choose your target server. This is the virtual recovery appliance on your ESX server.



Choose Target Server

Choose the Virtual Recovery Appliance that will provide protection for your servers.

▲ Current Servers

Servers:

		Server ▲	Version	Product
		BETA	8.1.0.613.0	DoubleTake Target

□ Show all servers

▼ Find a New Server

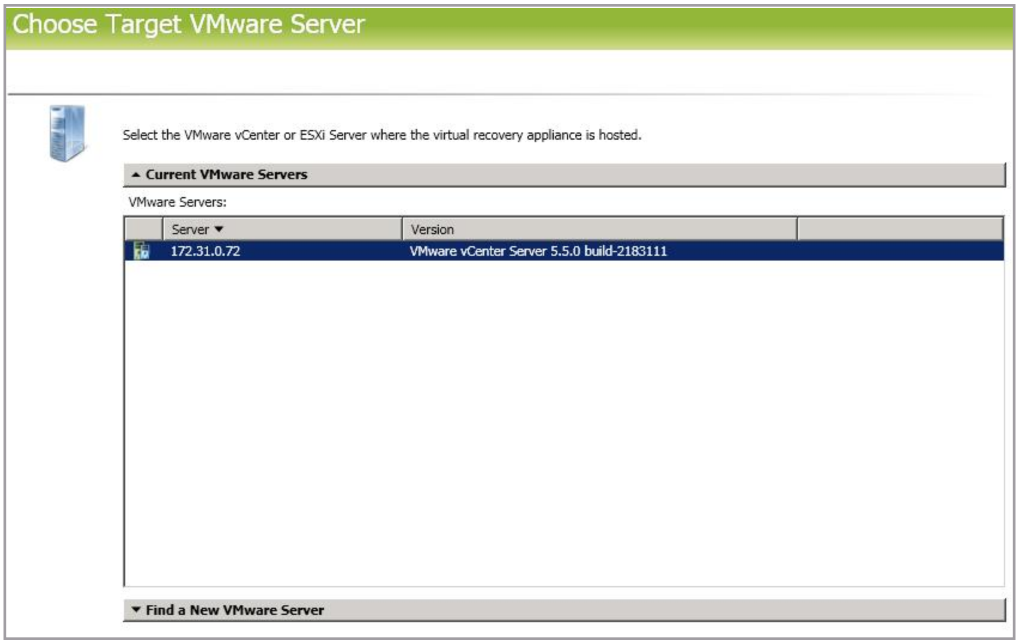
- **Current Servers**—This list contains the servers currently available in your console session. Servers that are not licensed for the workflow you have selected and those not applicable to the workload type you have selected will be filtered out of the list. Select your target server from the list. If the server you are looking for is not displayed, enable **Show all servers**. The servers in red are not available for the source server or workload type you have selected. Hover your mouse over an unavailable server to see a reason why this server is unavailable.

- **Find a New Server**—If the server you need is not in the **Current Servers** list, click the **Find a New Server** heading. From here, you can specify a server along with credentials for logging in to the server. If necessary, you can click **Browse** to select a server from a network drill-down list.

If you enter the target server's fully-qualified domain name, the Carbonite Replication Console will resolve the entry to the server short name. If that short name resides in two different domains, this could result in name resolution issues. In this case, enter the IP address of the server.

When specifying credentials for a new server, specify a user that is a member of the local dtadmin security group.

6. Click **Next** to continue.
7. Choose the server where your target virtual recovery appliance is located. This is also the server where your replica virtual machine will be located.



Choose Target VMware Server

Select the VMware vCenter or ESXi Server where the virtual recovery appliance is hosted.

Current VMware Servers

VMware Servers:

Server	Version
172.31.0.72	VMware vCenter Server 5.5.0 build-2183111

- **Current VMware Servers**—This list contains the vCenter and ESX servers currently available in your console session. Select your server from the list.
- **Find a New VMware Server**—If the server you need is not in the **Current VMware Servers** list, click the **Find a New VMware Server** heading.
    - **vCenter/ESXi Server**—Select your server from the list. If your server is not in the list, manually type it in.

- **User name**—Specify the root user or another user that has the administrator role on the specified server.
- **Password**—Specify the password associated with the **User name** you entered.
- **Domain**—If you are working in a domain environment, specify the **Domain**.

If your server name does not match the security certificate or the security certificate has expired, you will be prompted if you want to install the untrusted security certificate.

8. Click **Next** to continue.

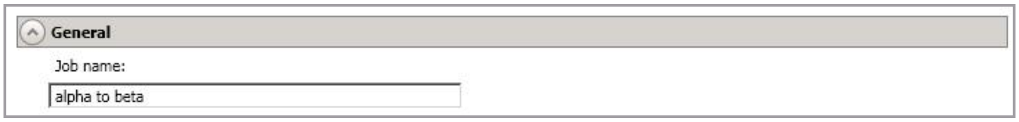
> You may be prompted for a route from the target to the source. This route is used so the target can communicate with the source to build job options. This dialog box will be displayed only if needed.

9. You have many options available for your full server to ESX job. Configure those options that are applicable to your environment.

   Go to each page identified below to see the options available for that section of the **Set Options** page. After you have configured your options, continue with the next step on page 187.

   - *General* on page 165
   - *Replica Virtual Machine Location* on page 166
   - *Replica Virtual Machine Configuration* on page 166
   - *Replica Virtual Machine Volumes* on page 168
   - *Replica Virtual Machine Network Settings* on page 175
   - *Test Failover* on page 176
   - *Failover Monitor* on page 179
   - *Failover Options* on page 181
   - *Mirror, Verify & Orphaned Files* on page 182
   - *Network Route* on page 184
   - *Snapshots* on page 185
   - *Compression* on page 186
   - *Bandwidth* on page 187

General

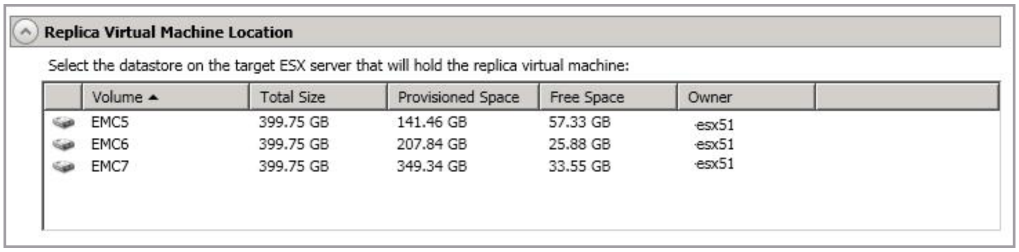


General

Job name: alpha to beta

For the **Job name**, specify a unique name for your job.

Replica Virtual Machine Location



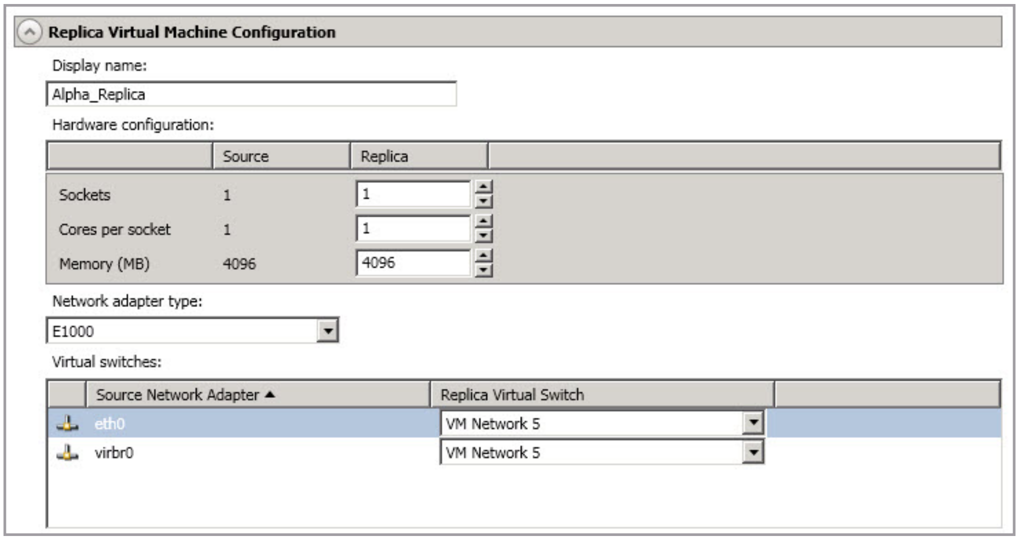
Replica Virtual Machine Location

Select the datastore on the target ESX server that will hold the replica virtual machine:

Volume ▲	Total Size	Provisioned Space	Free Space	Owner
EMC5	399.75 GB	141.46 GB	57.33 GB	esx51
EMC6	399.75 GB	207.84 GB	25.88 GB	esx51
EMC7	399.75 GB	349.34 GB	33.55 GB	esx51

Select one of the volumes from the list to indicate the volume on the target where you want to store the configuration files for the new virtual server when it is created. The target volume must have enough **Free Space**. You can select the location of the .vmdk files under **Replica Virtual Machine Volumes**.

Replica Virtual Machine Configuration



Replica Virtual Machine Configuration

Display name: Alpha_Replica

Hardware configuration:

	Source	Replica
Sockets	1	1
Cores per socket	1	1
Memory (MB)	4096	4096

Network adapter type: E1000

Virtual switches:

Source Network Adapter ▲	Replica Virtual Switch
eth0	VM Network 5
virbr0	VM Network 5

- **Display name**—Specify the name of the replica virtual machine. This will be the display name of the virtual machine on the host system.
- **Hardware configuration**—Specify how you want the replica virtual machine to be created.
  - **Sockets**—Specify how many sockets to create on the new virtual machine. The number of sockets on the source is displayed to guide you in making an appropriate selection. If you select fewer sockets than the source, your clients may be impacted by slower responses.
  - **Cores per socket**—Specify how many cores to create per socket. The number of cores per socket on the source is displayed to guide you in making an appropriate selection.

    - **Memory**—Specify the amount of memory, in MB, to create on the new virtual machine. The memory on the source is displayed to guide you in making an appropriate selection. If you select less memory than the source, your clients may be impacted by slower responses.
- **Network adapter type**—If your target appliance has VMware Tools installed, you can select the type of adapter, **E1000** or **VmxNet3**, to use on the replica virtual machine. This selection will apply to all adapters on the replica.

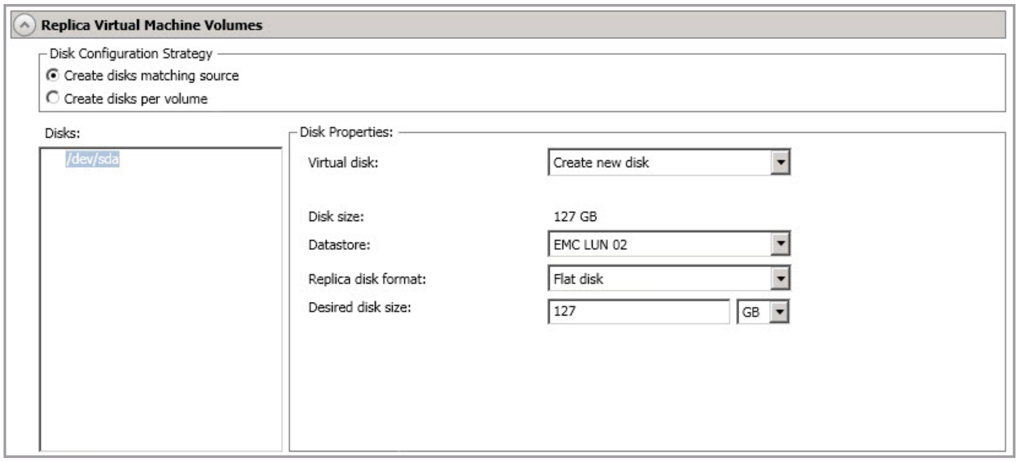
  If your source has VMware Tools installed, but it is an older version than VMware Tools installed on your target appliance, you will have to update VMware Tools on the replica server after failover in order to get the VmxNet3 adapter to function.

- **Virtual switches**—Identify how you want to handle the network mapping after failover. The **Source Network Adpater** column lists the NICs from the source. Map each one to a **Replica Virtual Switch**, which is a virtual network on the target. You can also choose to discard the source's NIC and IP addresses.

Replica Virtual Machine Volumes

- **Create disks matching source**—Select this option if you want the disk configuration on the target replica to match the disk configuration on the source.

> If your source has LVM, the logical volume groups on the source cannot contain physical volumes which are created based on unpartitioned disks, such as /dev/sdb. Instead, partitions should be created on the disks first, such as /dev/sdb1, and physical volumes should be created based on the partitions, before applying them to the logical volume groups. If your source physical volumes are based on unpartitioned disks, you must select the per volume configuration.



Replica Virtual Machine Volumes

Disk Configuration Strategy

● Create disks matching source

○ Create disks per volume

Disks: /dev/sda

Disk Properties:

Virtual disk:	Create new disk
Disk size:	127 GB
Datastore:	EMC LUN 02
Replica disk format:	Flat disk
Desired disk size:	127 GB

  - **Virtual Disk**—Specify if you want Carbonite Availability to create a new disk for your replica virtual machine or if you want to use an existing disk. If you have more than one disk, you cannot mix and match new and existing. They must all be new disks or all existing disks.

    Reusing a virtual disk can be useful for pre-staging data on a LAN and then relocating the virtual disk to a remote site after the initial mirror is complete. You save time by skipping the virtual disk creation steps and performing a difference mirror instead of a full mirror. With pre-staging, less data will need to be sent across the wire initially. In order to use an existing virtual disk, it must be a valid virtual disk, it cannot be attached to any other virtual machine, and it cannot have any associated snapshots.

    Each pre-existing disk must be located on the target datastore specified. If you have copied the .vmdk file to this location manually, be sure you have also copied the associated -flat.vmdk file too. If you have used vCenter to copy the virtual machine, the associated file will automatically be copied. There are no restrictions on the file name of the .vmdk, but the associated -flat.vmdk file must have the same base name and the reference to that flat file in the .vmdk must be correct. Carbonite Availability will move, not copy, the virtual disk files to the appropriate folders created by the replica, so make sure the selected target datastore is where you want the replica virtual disk to be located.

In a WAN environment, you may want to take advantage of using an existing disk by using a process similar to the following.

a. Create a job in a LAN environment, letting Carbonite Availability create the virtual disk for you.
b. Complete the mirror process locally.
c. Delete the job and when prompted, do not delete the replica.
d. Move the virtual disk files to the desired target datastore. Do not forget to move the associated -flat.vmdk file if you move the files manually.
e. Create a new protection job for the same source and reuse your existing disk.

If you have reused some existing disks and created some new disks, the numbering of the hard disks will not be identical on the source and the replica virtual machine. New disks will be created first and then existing disks will be attached. VMware assigns the hard disk numbers in order of creation and then those that are attached. The Virtual Device Node SCSI IDs will still be correct and there will be no impact within the guest of the replica virtual machine.

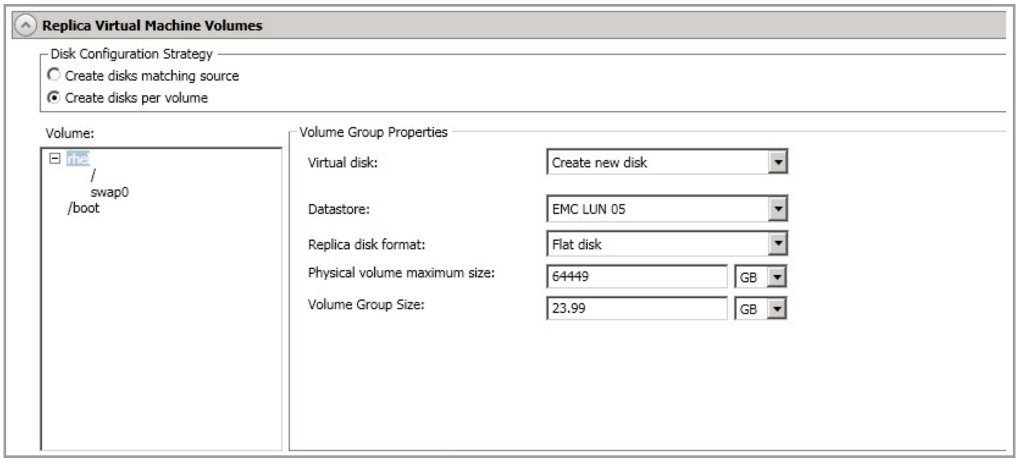
If your source has multiple partitions inside a single .vmdk, you can only use an existing virtual disk that Carbonite Availability created. You can only use an existing virtual disk created outside of Carbonite Availability if there is one partition in each pre-existing disk.

If you are using Logical Volume Manager, then you can only use existing disks when creating a new full server to ESX appliance job if the existing disks were created using Carbonite Availability version 7.1 or later. Versions prior to 7.1 have important LVM information deleted when the job is deleted, thus you cannot reuse the disk for a future job. If you are not using LVM, this is not an issue.

You cannot reuse a virtual disk if it has snapshots associated with it. You must delete all snapshots before you can reuse a virtual disk.

- **Datastore**—Specify the datastore where you want to store the .vmdk files for the disk. You can specify the location of the virtual machine configuration files in the **Replica Virtual Machine Location** section.
- **Replica disk format**—If you are creating a new disk, specify the format of the disk that will be created.
  - **Flat Disk**—This disk format allocates the full amount of the disk space immediately, but does not initialize the disk space to zero until it is needed.
  - **Thick**—This disk format allocates the full amount of the disk space immediately, initializing all of the allocated disk space to zero.
  - **Thin**—This disk format does not allocate the disk space until it is needed.
- **Desired disk size**—If you are creating a new disk, specify the maximum size, in MB or GB, of the disks.
- **Pre-existing disk path**—If you are using an existing virtual disk, specify the location of the existing virtual disk that you want to reuse.

- **Create disks per volume**—Select this option if you want the disk configuration on the target replica to be per source volume.
    - **Volume Group Properties**—If your source has volume groups, you will see them listed in the **Volume** list. Highlight a volume group and set the available **Volume Group Properties** that are displayed to the right of the **Volume** list. The fields displayed in the **Volume Group Properties** will depend on your selection for **Virtual disk**.



Replica Virtual Machine Volumes

Disk Configuration Strategy

○ Create disks matching source

● Create disks per volume

Volume:

```
⊟ rhe
      /
      swap0
  /boot
```

Volume Group Properties

Virtual disk: Create new disk

Datastore: EMC LUN 05

Replica disk format: Flat disk

Physical volume maximum size: 64449 GB

Volume Group Size: 23.99 GB

- **Virtual Disk**—Specify if you want Carbonite Availability to create a new disk for your replica virtual machine or if you want to use an existing disk.

  Reusing a virtual disk can be useful for pre-staging data on a LAN and then relocating the virtual disk to a remote site after the initial mirror is complete. You save time by skipping the virtual disk creation steps and performing a difference mirror instead of a full mirror. With pre-staging, less data will need to be sent across the wire initially. In order to use an existing virtual disk, it must be a valid virtual disk, it cannot be attached to any other virtual machine, and it cannot have any associated snapshots.

  Each pre-existing disk must be located on the target datastore specified. If you have copied the .vmdk file to this location manually, be sure you have also copied the associated -flat.vmdk file too. If you have used vCenter to copy the virtual machine, the associated file will automatically be copied. There are no restrictions on the file name of the .vmdk, but the associated -flat.vmdk file must have the same base name and the reference to that flat file in the .vmdk must be correct. Carbonite Availability will move, not copy, the virtual disk files to the appropriate folders created by the replica, so make sure the selected target datastore is where you want the replica virtual disk to be located.

  In a WAN environment, you may want to take advantage of using an existing disk by using a process similar to the following.

  a. Create a job in a LAN environment, letting Carbonite Availability create the virtual disk for you.
  b. Complete the mirror process locally.
  c. Delete the job and when prompted, do not delete the replica.

d. Move the virtual disk files to the desired target datastore. Do not forget to move the associated -flat.vmdk file if you move the files manually.

e. Create a new protection job for the same source and reuse your existing disk.

If you have reused some existing disks and created some new disks, the numbering of the hard disks will not be identical on the source and the replica virtual machine. New disks will be created first and then existing disks will be attached. VMware assigns the hard disk numbers in order of creation and then those that are attached. The Virtual Device Node SCSI IDs will still be correct and there will be no impact within the guest of the replica virtual machine.

If your source has multiple partitions inside a single .vmdk, you can only use an existing virtual disk that Carbonite Availability created. You can only use an existing virtual disk created outside of Carbonite Availability if there is one partition in each pre-existing disk.

If you are using Logical Volume Manager, then you can only use existing disks when creating a new full server to ESX appliance job if the existing disks were created using Carbonite Availability version 7.1 or later. Versions prior to 7.1 have important LVM information deleted when the job is deleted, thus you cannot reuse the disk for a future job. If you are not using LVM, this is not an issue.

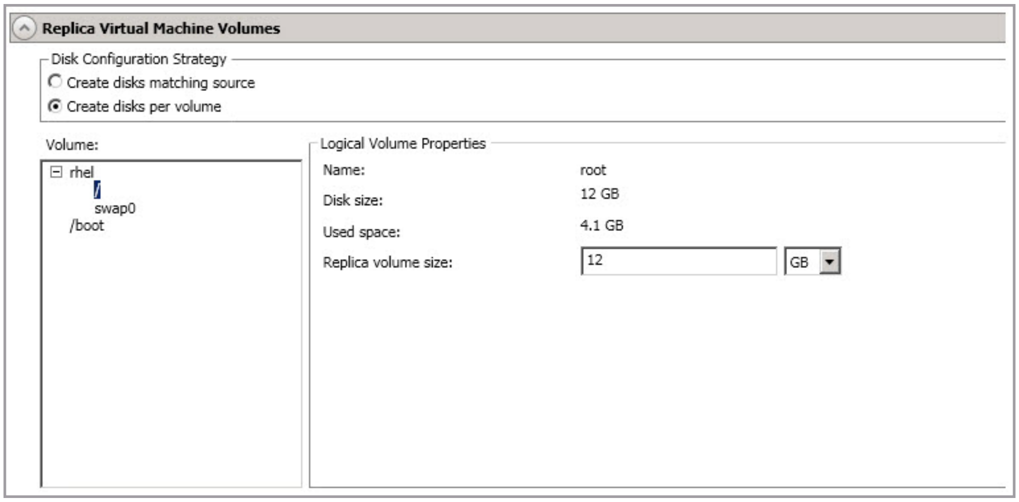
You cannot reuse a virtual disk if it has snapshots associated with it. You must delete all snapshots before you can reuse a virtual disk.

- **Datastore**—Specify the datastore where you want to store the .vmdk files for the volume group. You can specify the location of the virtual machine configuration files in the **Replica Virtual Machine Location** section.
- **Replica disk format**—If you are creating a new disk, specify the format of the disk that will be created.
  - **Flat Disk**—This disk format allocates the full amount of the disk space immediately, but does not initialize the disk space to zero until it is needed.
  - **Thick**—This disk format allocates the full amount of the disk space immediately, initializing all of the allocated disk space to zero.
  - **Thin**—This disk format does not allocate the disk space until it is needed.
- **Physical volume maximum size**—If you are creating a new disk, specify the maximum size, in MB or GB, of the virtual disks used to create the volume group. The default value is equal to the maximum size that can be attached to the datastore you selected. That will depend on your ESX version, your file system version, and the block size of your datastore.
- **Volume Group size**—If you are creating a new disk, specify the maximum size, in MB or GB, of the volume group. The default value will match the

source. This value cannot be less than the logical volumes total size that you are trying to create on the volume group.

- **Pre-existing virtual disks path**—If you are using an existing virtual disk, specify the location of the existing virtual disks that you want to reuse.

- **Logical Volume Properties**—If your source has logical volumes, you will see them listed in the **Volume** list. Highlight a logical volume and set the available **Logical Volume Properties** that are displayed to the right of the **Volume** list.



Replica Virtual Machine Volumes

Disk Configuration Strategy

○ Create disks matching source

● Create disks per volume

Volume:

⊟ rhel
- /
- swap0

/boot

Logical Volume Properties

Name: root

Disk size: 12 GB

Used space: 4.1 GB

Replica volume size: 12 GB

If you are using an existing virtual disk, you will not be able to modify the logical volume properties.

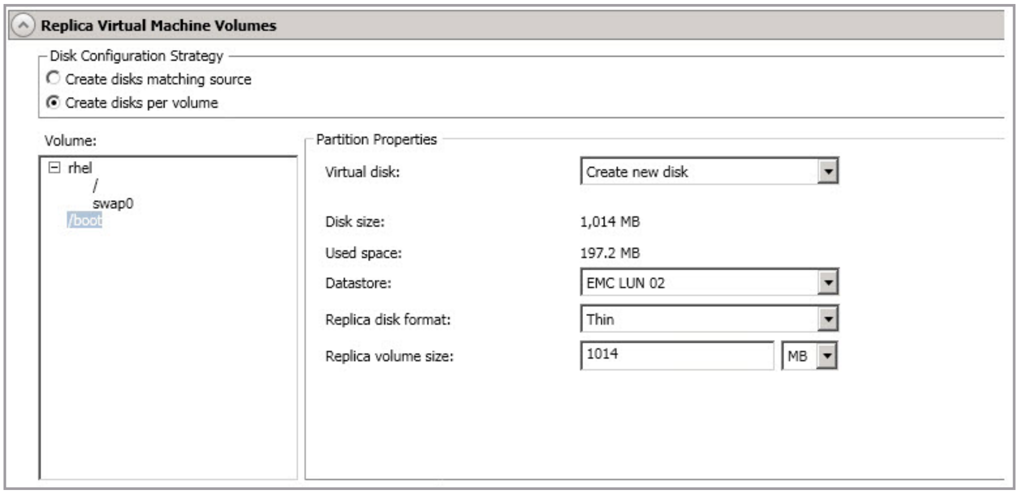
The size and space displayed may not match the output of the Linux df command. This is because df shows the size of the mounted file system not the underlying partition which may be larger. Additionally, Carbonite Availability uses powers of 1024 when computing GB, MB, and so on. The df command typically uses powers of 1000 and rounds up to the nearest whole value.

- **Name**—This field displays the logical volume name.
- **Disk size**—This field displays the size of the logical volume on the source.
- **Used space**—This field displays the amount of disk space in use on the source logical volume.
- **Replica volume size**—Specify the size, in MB or GB, of the replica logical volume on the target. The value must be at least the size of the specified **Used space** on that volume.

In some cases, the replica virtual machine may use more virtual disk space than the size of the source volume due to differences in how the virtual disk's block size is formatted and how hard links are handled.

To avoid this issue, specify the size of your replica to be at least 5 GB larger.

- **Partition Properties**—If your source has partitions, you will see them listed in the **Volume** list. Highlight a partition and set the available **Partition Properties** that are displayed to the right of the **Volume** list. The fields displayed in the **Partition Properties** will depend on your selection for **Virtual disk**.



Replica Virtual Machine Volumes

Disk Configuration Strategy

○ Create disks matching source

● Create disks per volume

Volume:

```
⊟ rhel
    /
    swap0
  /boot
```

Partition Properties

Virtual disk: Create new disk

Disk size: 1,014 MB

Used space: 197.2 MB

Datastore: EMC LUN 02

Replica disk format: Thin

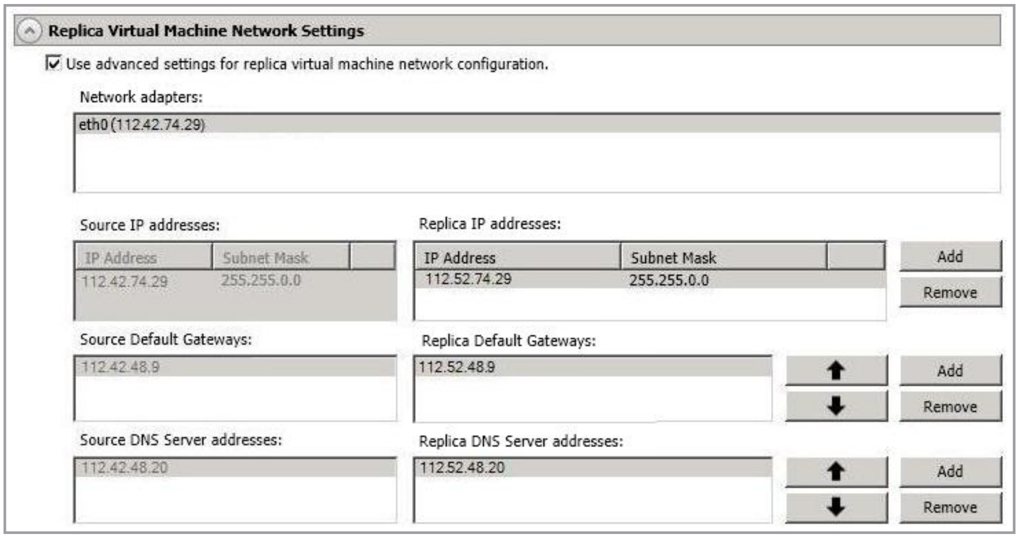
Replica volume size: 1014 MB

The size and space displayed may not match the output of the Linux df command. This is because df shows the size of the mounted file system not the underlying partition which may be larger. Additionally, Carbonite Availability uses powers of 1024 when computing GB, MB, and so on. The df command typically uses powers of 1000 and rounds up to the nearest whole value.

  - **Virtual Disk**—Specify if you want Carbonite Availability to create a new disk for your replica virtual machine or if you want to use an existing disk. Review the details above under **Volume Group Properties Virtual Disk** for information on using an existing disk.
  - **Disk size**—This field displays the size of the partition on the source.
  - **Used space**—This field displays the amount of disk space in use on the source partition.
  - **Datastore**—Specify the datastore where you want to store the .vmdk files for the partition. You can specify the location of the virtual machine configuration files in the **Replica Virtual Machine Location** section.
  - **Replica disk format**—Specify the format of the disk that will be created.
    - **Flat Disk**—This disk format allocates the full amount of the disk space immediately, but does not initialize the disk space to zero until it is needed.

  - **Thick**—This disk format allocates the full amount of the disk space immediately, initializing all of the allocated disk space to zero.
  - **Thin**—This disk format does not allocate the disk space until it is needed.
- **Replica volume size**—Specify the size, in MB or GB, of the replica partition on the target. The value must be at least the size of the specified **Used space** on that partition.
- **Pre-existing disks path**—If you are using an existing virtual disk, specify the location of the existing virtual disks that you want to reuse.

Replica Virtual Machine Network Settings



Replica Virtual Machine Network Settings

☑ Use advanced settings for replica virtual machine network configuration.

Network adapters:

eth0 (112.42.74.29)

Source IP addresses:

IP Address	Subnet Mask
112.42.74.29	255.255.0.0

Replica IP addresses:

IP Address	Subnet Mask
112.52.74.29	255.255.0.0

Source Default Gateways: 112.42.48.9

Replica Default Gateways: 112.52.48.9

Source DNS Server addresses: 112.42.48.20

Replica DNS Server addresses: 112.52.48.20

- **Use advanced settings for replica virtual machine network configuration**—Select this option to enable the replica virtual machine network setting configuration. This setting is primarily used for WAN support.
- **Network adapters**—Select a network adapter from the source and specify the **Replica IP addresses**, **Replica Default Gateways**, and **Replica DNS Server addresses** to be used after failover. If you add multiple gateways or DNS servers, you can sort them by using the arrow up and arrow down buttons. Repeat this step for each network adapter on the source.

Updates made during failover will be based on the network adapter name when protection is established. If you change that name, you will need to delete the job and re-create it so the new name will be used during failover.

If you update one of the advanced settings (IP address, gateway, or DNS server), then you must update all of them. Otherwise, the remaining items will be left blank. If you do not specify any of the advanced settings, the replica virtual machine will be assigned the same network configuration as the source.

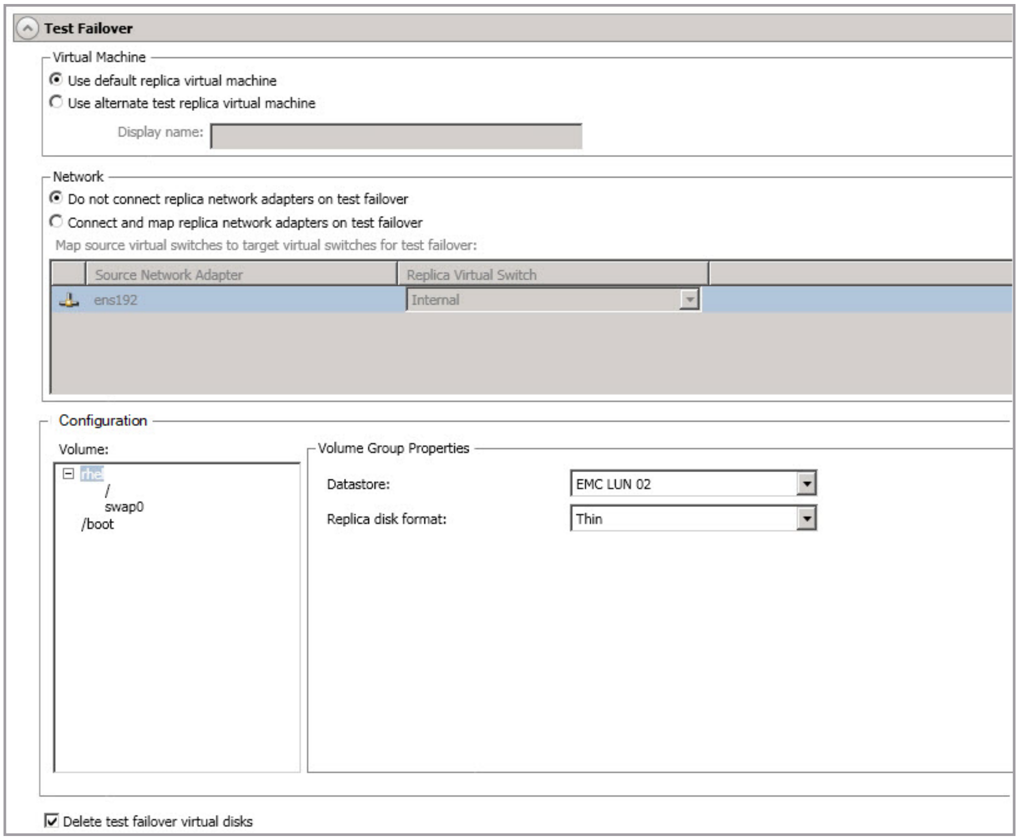
By default, the source IP address will be included in the target IP address list as the default address. If you do not want the source IP address to be the default address on the target after failover, remove that address from the **Replica IP addresses** list.

Linux operating systems only support one gateway, so the first gateway listed will be used.

Test Failover

These options allow you to perform a test failover. Keep in mind the following for using test failover.

- All data volumes must be under LVM for test failover.
- In order to use test failover, you must make sure your target has sufficient free disk space available. The free space on each volume group on the target must be larger than 50% of the total size of all logical volumes in that volume group on the source. Meeting that amount of free space may depend on the **Disk Configuration Strategy** you selected under **Replica Virtual Disk Volumes**.
  - **Create disks matching source**—With the match source option, you must have sufficient free space on the source before the job is created because the target disks will be matching the source. You may need to increase free disk space on the source (perhaps add a partition which has been created on the raw disk on the source and then extend the volume group), in order to allow for sufficient free space to be matched on the target.
  - **Create disks per volume**—With the per volume option, select a volume group size on the target that has enough free space to accommodate the 50% free space requirement.
- Test failover is not supported for Btrfs file systems.
- The source, target, and protection job will remain online and uninterrupted during the test.
- The test will be performed using the test failover settings configured during job creation.
- The test will use the current data on the target.
- Scheduled snapshots will be deferred during the test and taken automatically after the test is undone.
- The test failover will take a snapshot of the current data on the target and create a new set of virtual disks. The data from the snapshot will be mirrored to the new set of disks using the same mirroring options as the protection job.
- Once the mirror is complete, the replica virtual machine or alternate replica virtual machine, depending on your selected configuration, is automatically brought online using the new set of disks.
- The replica virtual machine or alternate replica virtual machine, depending on your selected configuration, will use the network settings specified in the test failover settings of the protection job.
- When you are finished with your test, undo it.
- When you undo a test failover, the new set of disks will be maintained or deleted as specified in the test failover settings of the protection job.
- At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)



Test Failover

Virtual Machine

● Use default replica virtual machine

○ Use alternate test replica virtual machine

Display name:

Network

● Do not connect replica network adapters on test failover

○ Connect and map replica network adapters on test failover

Map source virtual switches to target virtual switches for test failover:

Source Network Adapter	Replica Virtual Switch
ens192	Internal

Configuration

Volume:

- rhel
  - /
  - swap0
- /boot

Volume Group Properties

Datastore: EMC LUN 02

Replica disk format: Thin

☑ Delete test failover virtual disks

- **Use default replica virtual machine—**Select this option to use the same replica virtual machine that will be used for live failover for the test failover. Do not use this option if you are using the snapshot functionality on the target. If there are snapshots on the target and you are using the default replica virtual machine for test failover, the test will fail and an error message will be logged.
- **Use alternate test replica virtual machine**—Select this option to create an alternate replica virtual machine to use for the test failover. You must use this option if you want to also use the snapshot functionality on the target.
- **Display name**—Specify the name of the alternate replica virtual machine to use for the test. This will be the display name of the virtual machine on the host system.
- **Do not connect replica network adapters on test failover—**Select this option if you do not want the replica virtual machine used for the test to be connected to the network.
- **Connect and map replica network adapters on test failover**—Select this option if you want the replica virtual machine used for the test to be connected to the network. You will need to map each **Source Network Adapter** to a **Target Virtual Switch** for the test. You can also choose to discard the source's NIC and IP addresses.
- **Configuration**—The **Disk Configuration Strategy** you selected in the **Replica Virtual Machine Volumes** section will be used for your test failover. However, you can select different locations and disk formats, if desired.
  - **Datastore**—Specify the datastore where you want to store the selected volume or disk. This selection will only be used for a test failover.
  - **Replica disk format—**Specify the format of the disk that will be created during a

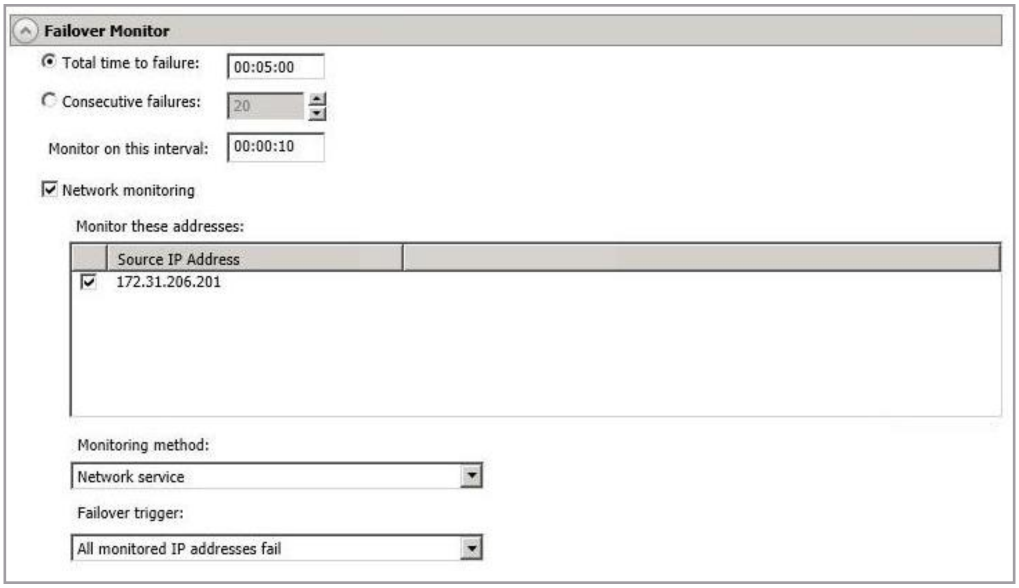
test failover.

- **Flat Disk**—This disk format allocates the full amount of the disk space immediately, but does not initialize the disk space to zero until it is needed.
- **Thick**—This disk format allocates the full amount of the disk space immediately, initializing all of the allocated disk space to zero.
- **Thin**—This disk format does not allocate the disk space until it is needed.

- **Delete test failover virtual disks**—Select this option if you want to delete the new virtual disks created during the test failover process. If you disable this option, the new disks will not be deleted when you perform undo failover. This option will not be available if you have selected to use an alternate test replica virtual machine. In this case, the disks will automatically be deleted.

Be careful if you choose to connect the network adapters for a test failover. Depending on your network adapter mappings, users may be able to access the target. Also, since the source is still online, there is a chance users may split between accessing the source or target.

Failover Monitor



Failover Monitor

● Total time to failure: 00:05:00

○ Consecutive failures: 20

Monitor on this interval: 00:00:10

☑ Network monitoring

Monitor these addresses:

	Source IP Address	
☑	172.31.206.201	

Monitoring method: Network service

Failover trigger: All monitored IP addresses fail

- **Total time to failure**—Specify, in hours:minutes:seconds, how long the target will keep trying to contact the source before the source is considered failed. This time is precise. If the total time has expired without a successful response from the source, this will be considered a failure.

  Consider a shorter amount of time for servers, such as a web server or order processing database, which must remain available and responsive at all times. Shorter times should be used where redundant interfaces and high-speed, reliable network links are available to prevent the false detection of failure. If the hardware does not support reliable communications, shorter times can lead to premature failover. Consider a longer amount of time for machines on slower networks or on a server that is not transaction critical. For example, failover would not be necessary in the case of a server restart.

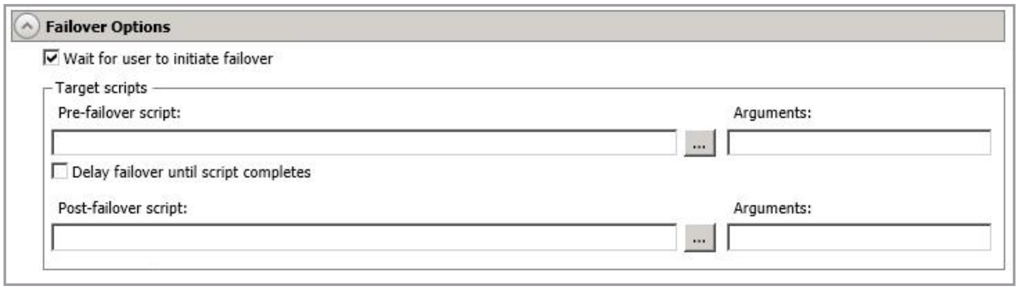
- **Consecutive failures—**Specify how many attempts the target will make to contact the source before the source is considered failed. For example, if you have this option set to 20, and your source fails to respond to the target 20 times in a row , this will be considered a failure.
- **Monitor on this interval**—Specify, in hours:minutes:seconds, how long to wait between attempts to contact the source to confirm it is online. This means that after a response (success or failure) is received from the source, Carbonite Availability will wait the specified interval time before contacting the source again. If you set the interval to 00:00:00, then a new check will be initiated immediately after the response is received.

  If you choose **Total time to failure**, do not specify a longer interval than failure time or your server will be considered failed during the interval period.

  If you choose **Consecutive failures**, your failure time is calculated by the length of time it takes your source to respond plus the interval time between each response, times the number of consecutive failures that can be allowed. That would be (response time + interval) * failure number. Keep in mind that timeouts from a failed check are included in the response time, so your failure time will not be precise.

- **Network monitoring**—With this option, the target will monitor the source using a network ping.
    - **Monitor these addresses**—Select each **Source IP Address** that you want the target to monitor. If you are using a NAT environment, do not select a private IP address on the source because the target cannot reach the source's private address, thus causing an immediate failure.
    - **Monitoring method**—This option determines the type of network ping used for failover monitoring.
        - **Network service**—Source availability will be tested by an ICMP ping to confirm the route is active.
        - **Replication service**—Source availability will be tested by a UDP ping to confirm the Double-Take service is active.
        - **Network and replication services**—Source availability will be tested by both an ICMP ping to confirm the route is active and a UDP ping to confirm the Double-Take service is active. If either monitoring method fails, failover will be triggered.
    - **Failover trigger**—If you are monitoring multiple IP addresses, specify when you want a failover condition to be triggered.
        - **One monitored IP address fails**—A failover condition will be triggered when any one of the monitored IP addresses fails. If each IP address is on a different subnet, you may want to trigger failover after one fails.
        - **All monitored IP addresses fail**—A failover condition will be triggered when all monitored IP addresses fail. If there are multiple, redundant paths to a server, losing one probably means an isolated network problem and you should wait for all IP addresses to fail.

Failover Options



Failover Options

☑ Wait for user to initiate failover

Target scripts

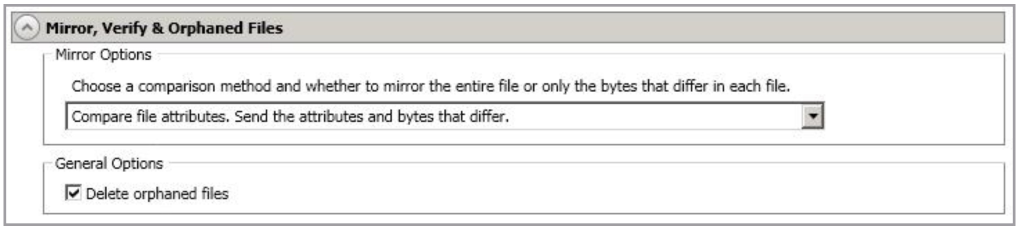
Pre-failover script: Arguments:

□ Delay failover until script completes

Post-failover script: Arguments:

- **Wait for user to initiate failover**—The failover process can wait for you to initiate it, allowing you to control when failover occurs. When a failure occurs, the job will wait in **Failover Condition Met** for you to manually initiate the failover process. Disable this option if you want failover to occur immediately when a failure occurs.
- **Target Scripts**—You can customize failover by running scripts on the target appliance or the replica. Scripts may contain any valid Linux command, executable, or shell script file. The scripts are processed using the same account running the Double-Take Management service. Examples of functions specified in scripts include stopping and starting services, stopping and starting applications or processes, notifying the administrator before and after failover occurs, and so on. There are two types of failover scripts.
    - **Pre-failover script**—This script runs on the target appliance at the beginning of the failover process. Specify the full path and name of the script file.
    - **Delay until script completes**—Enable this option if you want to delay the failover process until the associated script has completed. If you select this option, make sure your script handles errors, otherwise the failover process may never complete if the process is waiting on a script that cannot complete.
    - **Post-failover script**—This script runs on the replica at the end of the failover process. Specify the full path and name of the script file.
    - **Arguments**—Specify a comma-separated list of valid arguments required to execute the script.

Mirror, Verify & Orphaned Files



Mirror, Verify & Orphaned Files

Mirror Options

Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.

Compare file attributes. Send the attributes and bytes that differ.

General Options

☑ Delete orphaned files

- **Mirror Options**—Choose a comparison method and whether to mirror the entire file or only the bytes that differ in each file.
  - **Do not compare files. Send the entire file.**—Carbonite Availability will not perform any comparisons between the files on the source and target. All files will be mirrored to the target, sending the entire file. This option requires no time for comparison, but it can be slower because it sends the entire file. However, it is useful for configurations that have large data sets with millions of small files that are frequently changing and it is more efficient to send the entire file. You may also need to use this option if configuration management policies require sending the entire file.
  - **Compare file attributes. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and will mirror only the attributes and bytes that are different. This option is the fastest comparison method and fastest mirror option. Files that have not changed can be easily skipped. Also files that are open and require a checksum mirror can be compared.
  - **Compare file attributes and data. Send the attributes and bytes that differ.**—Carbonite Availability will compare file attributes and the file data and will mirror only the attributes and bytes that are different. This comparison method is not as fast because every file is compared, regardless of whether the file has changed or is open. However, sending only the attributes and bytes that differ is the fasted mirror option.

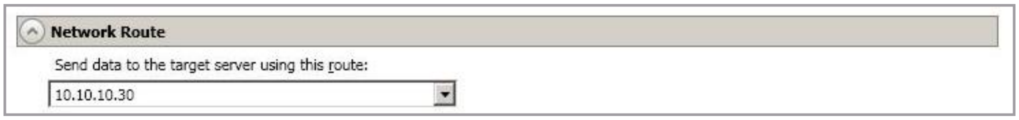
  If a file is small enough that mirroring the entire file is faster than comparing it and then mirroring it, Carbonite Availability will automatically mirror the entire file.

- **General Options**—Choose your general mirroring options.
  - **Delete orphaned files**—An orphaned file is a file that exists in the replica data on the target, but does not exist in the protected data on the source. This option specifies if orphaned files should be deleted on the target.

    Orphaned file configuration is a per target configuration. All jobs to the same target will have the same orphaned file configuration.

    If delete orphaned files is enabled, carefully review any replication rules that use wildcard definitions. If you have specified wildcards to be excluded from protection, files matching those wildcards will also be excluded from orphaned file processing and will not be deleted from the target. However, if you have specified wildcards to be included in your protection, those files that fall outside the wildcard inclusion rule will be considered orphaned files and will be deleted from the target.

Network Route

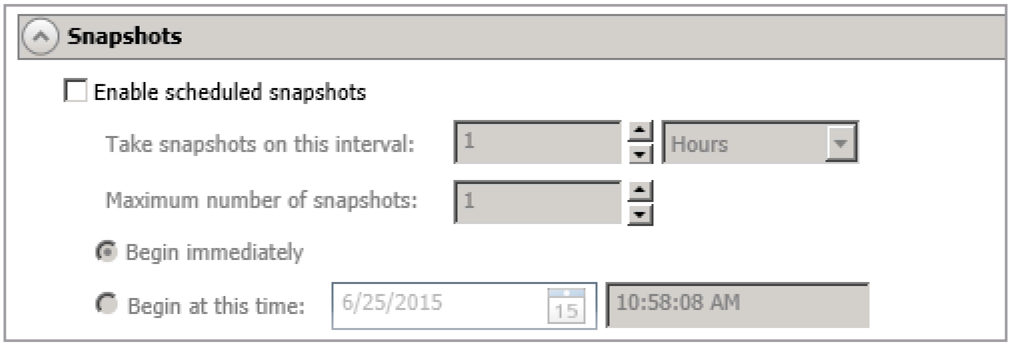

Network Route

Send data to the target server using this route:

10.10.10.30

By default, Carbonite Availability will select a target route for transmissions. If desired, specify an alternate route on the target that the data will be transmitted through. This allows you to select a different route for Carbonite Availability traffic. For example, you can separate regular network traffic and Carbonite Availability traffic on a machine with multiple IP addresses. You can also select or manually enter a public IP address (which is the public IP address of the server's router) if you are using a NAT environment.

Snapshots



Snapshots

□ Enable scheduled snapshots

Take snapshots on this interval: 1 Hours

Maximum number of snapshots: 1

● Begin immediately

○ Begin at this time: 6/25/2015 10:58:08 AM

A snapshot is an image of the source replica data on the target taken at a single point in time. You can failover to a snapshot. However, you cannot access the snapshot to recover specific files or folders.

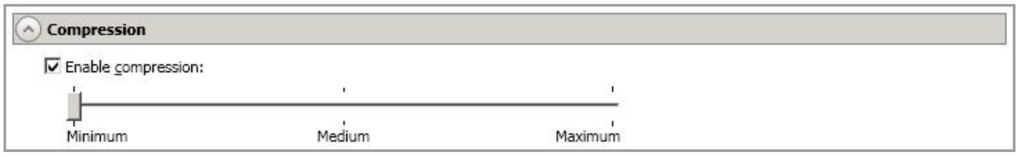
Turn on **Enable scheduled snapshots** if you want Carbonite Availability to take snapshots automatically at set intervals.

- **Take snapshots on this interval**—Specify the interval (in days, hours, or minutes) for taking snapshots. Due to VMware processing speed, you should not schedule at less than 10 minute intervals.
- **Maximum number of snapshots**—Specify the maximum number of snapshots to retain. The upper limit is 30 maximum snapshots. Once this limit is reached, a new snapshot will delete the oldest snapshot.
- **Begin immediately**—Select this option if you want to start taking snapshots immediately after the protection job is established.
- **Begin at this time**—Select this option if you want to start taking snapshots at a later date and time. Specify the date and time parameters to indicate when you want to start.

See *Managing snapshots* on page 59 for details on taking manual snapshots and deleting snapshots.

Make sure you have reviewed the snapshots best practices as noted in the *Full server to ESX requirements* on page 154.

Compression



Compression

☑ Enable compression:

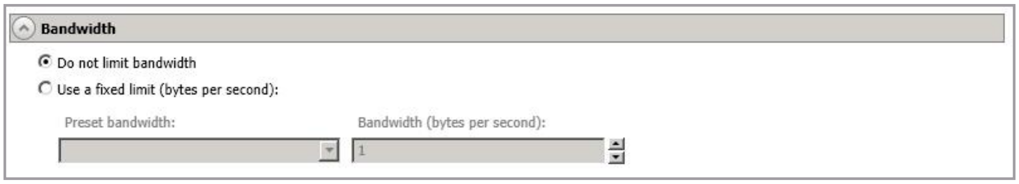
To help reduce the amount of bandwidth needed to transmit Carbonite Availability data, compression allows you to compress data prior to transmitting it across the network. In a WAN environment this provides optimal use of your network resources. If compression is enabled, the data is compressed before it is transmitted from the source. When the target receives the compressed data, it decompresses it and then writes it to disk. You can set the level from **Minimum** to **Maximum** to suit your needs.

Keep in mind that the process of compressing data impacts processor usage on the source. If you notice an impact on performance while compression is enabled in your environment, either adjust to a lower level of compression, or leave compression disabled. Use the following guidelines to determine whether you should enable compression.

- If data is being queued on the source at any time, consider enabling compression.
- If the server CPU utilization is averaging over 85%, be cautious about enabling compression.
- The higher the level of compression, the higher the CPU utilization will be.
- Do not enable compression if most of the data is inherently compressed. Many image (.jpg, .gif) and media (.wmv, .mp3, .mpg) files, for example, are already compressed. Some images files, such as .bmp and .tif, are decompressed, so enabling compression would be beneficial for those types.
- Compression may improve performance even in high-bandwidth environments.
- Do not enable compression in conjunction with a WAN Accelerator. Use one or the other to compress Carbonite Availability data.

All jobs from a single source connected to the same IP address on a target will share the same compression configuration.

Bandwidth



Bandwidth

◉ Do not limit bandwidth

○ Use a fixed limit (bytes per second):

Preset bandwidth:

Bandwidth (bytes per second): 1

Bandwidth limitations are available to restrict the amount of network bandwidth used for Carbonite Availability data transmissions. When a bandwidth limit is specified, Carbonite Availability never exceeds that allotted amount. The bandwidth not in use by Carbonite Availability is available for all other network traffic.

All jobs from a single source connected to the same IP address on a target will share the same bandwidth configuration.

- **Do not limit bandwidth**—Carbonite Availability will transmit data using 100% bandwidth availability.
- **Use a fixed limit**—Carbonite Availability will transmit data using a limited, fixed bandwidth. Select a **Preset bandwidth** limit rate from the common bandwidth limit values. The **Bandwidth** field will automatically update to the bytes per second value for your selected bandwidth. This is the maximum amount of data that will be transmitted per second. If desired, modify the bandwidth using a bytes per second value. The minimum limit should be 3500 bytes per second.

10. Click **Next** to continue.
11. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

    Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

    If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

    After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Migrate Reference Guide* for details on the various Carbonite Availability log files.
12. Once your servers have passed validation and you are ready to establish protection, click **Finish**,

and you will automatically be taken to the **Jobs** page.

Jobs in a NAT environment may take longer to start.

Once a job is created, do not change the name of underlying hardware components used in the job. For example, volume and datastore names or network adapter and virtual switch names. Any component used by name in your job must continue to use that name throughout the lifetime of job. If you must change a name, you will need to delete the job and re-create it using the new component name.

Managing and controlling full server to ESX jobs

Click **Jobs** from the main Carbonite Replication Console toolbar. The **Jobs** page allows you to view status information about your jobs. You can also control your jobs from this page.

The jobs displayed in the right pane depend on the server group folder selected in the left pane. Every job for each server in your console session is displayed when the **Jobs on All Servers** group is selected. If you have created and populated server groups (see *Managing servers* on page 24), then only the jobs associated with the server or target servers in that server group will be displayed in the right pane.

- *Overview job information displayed in the top right pane* on page 189
- *Detailed job information displayed in the bottom right pane* on page 192
- *Job controls* on page 194

Overview job information displayed in the top right pane

The top pane displays high-level overview information about your jobs. You can sort the data within a column in ascending and descending order. You can also move the columns to the left or right of each other to create your desired column order. The list below shows the columns in their default left to right order.

If you are using server groups, you can filter the jobs displayed in the top right pane by expanding the **Server Groups** heading and selecting a server group.

Column 1 (Blank)

The first blank column indicates the state of the job.

✅ A green circle with a white checkmark indicates the job is in a healthy state. No action is required.

⚠ A yellow triangle with a black exclamation point indicates the job is in a pending or warning state. This icon is also displayed on any server groups that you have created that contain a job in a pending or warning state. Carbonite Availability is working or waiting on a pending process or attempting to resolve the warning state.

❌ A red circle with a white X indicates the job is in an error state. This icon is also displayed on any server groups that you have created that contain a job in an error state. You will need to investigate and resolve the error.

? The job is in an unknown state.

Job

The name of the job

Source Server

The name of the source. This could be the name or IP address of your source.

Target Server

The name of the target. This could be the name or IP address of your target.

Job Type

Each job type has a unique job type name. This job is a Full Server to ESX job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the job details. Keep in mind that **Idle** indicates console to server activity is idle, not that your servers are idle.

Mirror Status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.
- **Idle**—Data is not being mirrored.
- **Paused**—Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans**—Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown**—The console cannot determine the status.

Replication Status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory**—Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown**—The console cannot determine the status.

Transmit Mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Operating System

The job type operating system

Detailed job information displayed in the bottom right pane

The details displayed in the bottom pane provide additional information for the job highlighted in the top pane. You can expand or collapse the bottom pane by clicking on the **Job Highlights** heading.

Name

The name of the job

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready—**The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the recovery point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the recovery point latency time.

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent (compressed)

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Connected since

The date and time indicating when the current job was started.

Recent activity

Displays the most recent activity for the selected job, along with an icon indicating the success or failure of the last initiated activity. Click the link to see a list of recent activities for the selected job. You can highlight an activity in the list to display additional details about the activity.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Job controls

You can control your job through the toolbar buttons available on the **Jobs** page. If you select multiple jobs, some of the controls will apply only to the first selected job, while others will apply to all of the selected jobs. For example, **View Job Details** will only show details for the first selected job, while **Stop** will stop protection for all of the selected jobs.

If you want to control just one job, you can also right click on that job and access the controls from the pop-up menu.

View Job Details

This button leaves the **Jobs** page and opens the **View Job Details** page.

Edit Job Properties

This button leaves the **Jobs** page and opens the **EditJob Properties** page.

Delete

Stops (if running) and deletes the selected jobs.

If you no longer want to protect the source and no longer need the replica of the source on the target, select to delete the associated replica virtual machine. Selecting this option will remove the job and completely delete the replica virtual machine on the target. Do not select this option if you want to keep the replica of the source on the target. If you do not select the delete option, the source replica will be preserved on the target.

If you are using vCenter, but created a job directly to an ESX host, you will have an orphaned virtual machine in vCenter if you choose to delete the virtual machine. That is because the ESX host is not forwarding the delete to the vCenter. You will need to manually delete the orphaned virtual machine in vCenter.

Provide Credentials

Changes the login credentials that the job (which is on the target machine) uses to authenticate to the servers in the job. This button opens the Provide Credentials dialog box where you can specify the new account information and which servers you want to update.

View Recent Activity

Displays the recent activity list for the selected job. Highlight an activity in the list to display additional details about the activity.

Start

Starts or resumes the selected jobs.

If you have previously stopped protection, the job will restart mirroring and replication.

If you have previously paused protection, the job will continue mirroring and replication from where it left off, as long as the Carbonite Availability queue was not exhausted during the time the job was paused. If the Carbonite Availability queue was exhausted during the time the job was paused, the job will restart mirroring and replication.

Also if you have previously paused protection, all jobs from the same source to the same IP address on the target will be resumed.

Pause

Pauses the selected jobs. Data will be queued on the source while the job is paused. Failover monitoring will continue while the job is paused.

All jobs from the same source to the same IP address on the target will be paused.

Stop

Stops the selected jobs. The jobs remain available in the console, but there will be no mirroring or replication data transmitted from the source to the target. Mirroring and replication data will not be queued on the source while the job is stopped, requiring a remirror when the job is restarted. The type of remirror will depend on your job settings. Failover monitoring will continue while the job is stopped.

Take Snapshot

Even if you have scheduled the snapshot process, you can run it manually any time. If an automatic or scheduled snapshot is currently in progress, Carbonite Availability will wait until that one is finished before taking the manual snapshot.

Manage Snapshots

Allows you to manage your snapshots by taking and deleting snapshots for the selected job. See *Managing snapshots* on page 59 for more information.

Failover or Cutover

Starts the failover process. See *Failing over full server to ESX jobs* on page 206 for the process and details of failing over a full server to ESX job.

Failback

Starts the failback process. Failback does not apply to full server to ESX jobs.

Restore

Starts the restoration process. Restore does not apply to full server to ESX jobs.

Reverse

Reverses protection. Reverse protection does not apply to full server to ESX jobs.

Undo Failover or Cutover

Cancels a test failover by undoing it. Undo failover does not apply to full server to ESX jobs.

View Job Log

Opens the job log. On the right-click menu, this option is called **View Logs**, and you have the option of opening the job log, source server log, or target server log.

Other Job Actions

Opens a small menu of other job actions. These job actions are not available for Linux jobs.

Generate Activity Report

For all Windows jobs except files and folders, you can create a basic failover report. This is the same report created by Get-DtLatestFailoverReport and Get-DtAllFailoverReports from the Carbonite Availability PowerShell module. The report will be located on the target in the \Service\Reports directory where Carbonite Availability is installed. For Linux jobs, you must use PowerShell.

Filter

Select a filter option from the drop-down list to only display certain jobs. You can display **Healthy jobs**, **Jobs with warnings**, or **Jobs with errors**. To clear the filter, select **All jobs**. If you have created and populated server groups, then the filter will only apply to the jobs associated with the server or target servers in that server group. See *Managing servers* on page 24.

Search

Allows you to search the source or target server name for items in the list that match the criteria you have entered.

Overflow Chevron

Displays any toolbar buttons that are hidden from view when the window size is reduced.

Viewing full server to ESX job details

From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.

Review the following table to understand the detailed information about your job displayed on the **View Job Details** page.

Job name

The name of the job

Job type

Each job type has a unique job type name. This job is a Full Server to ESX job. For a complete list of all job type names, press F1 to view the Carbonite Replication Console online help.

Health

✓ The job is in a healthy state.

⚠ The job is in a warning state.

✖ The job is in an error state.

? The job is in an unknown state.

Activity

There are many different **Activity** messages that keep you informed of the job activity. Most of the activity messages are informational and do not require any administrator interaction. If you see error messages, check the rest of the job details.

Connection ID

The incremental counter used to number connections. The number is incremented when a connection is created. The counter is reset if there are no existing jobs and the Double-Take service is restarted.

Transmit mode

- **Active**—Data is being transmitted to the target.
- **Paused**—Data transmission has been paused.
- **Scheduled**—Data transmission is waiting on schedule criteria.
- **Stopped**—Data is not being transmitted to the target.
- **Error**—There is a transmission error.
- **Unknown**—The console cannot determine the status.

Target data state

- **OK**—The data on the target is in a good state.
- **Mirroring**—The target is in the middle of a mirror process. The data will not be in a good state until the mirror is complete.
- **Mirror Required**—The data on the target is not in a good state because a remirror is required. This may be caused by an incomplete or stopped mirror or an operation may have been dropped on the target.
- **Busy**—The source is low on memory causing a delay in getting the state of the data on the target.
- **Not Loaded**—Carbonite Availability target functionality is not loaded on the target server. This may be caused by a license key error.
- **Not Ready**—The Linux drivers have not yet completed loading on the target.
- **Unknown**—The console cannot determine the status.

Target route

The IP address on the target used for Carbonite Availability transmissions.

Compression

- **On** / **Level**—Data is compressed at the level specified.
- **Off**—Data is not compressed.

Encryption

- **On**—Data is being encrypted before it is sent from the source to the target.
- **Off**—Data is not being encrypted before it is sent from the source to the target.

Bandwidth limit

If bandwidth limiting has been set, this statistic identifies the limit. The keyword **Unlimited** means there is no bandwidth limit set for the job.

Connected since

The source server date and time indicating when the current job was started. This field is blank, indicating that a TCP/IP socket is not present, when the job is waiting on transmit options or if the transmission has been stopped. This field will maintain the date and time, indicating that a TCP/IP socket is present, when transmission has been paused.

Additional information

Depending on the current state of your job, you may see additional information displayed to keep you informed about the progress and status of your job. If there is no additional information, you will see (None) displayed.

Mirror status

- **Calculating**—The amount of data to be mirrored is being calculated.
- **In Progress**—Data is currently being mirrored.
- **Waiting**—Mirroring is complete, but data is still being written to the target.

- **Idle**—Data is not being mirrored.
- **Paused—**Mirroring has been paused.
- **Stopped**—Mirroring has been stopped.
- **Removing Orphans—**Orphan files on the target are being removed or deleted depending on the configuration.
- **Verifying**—Data is being verified between the source and target.
- **Unknown—**The console cannot determine the status.

Mirror percent complete

The percentage of the mirror that has been completed

Mirror remaining

The total number of mirror bytes that are remaining to be sent from the source to the target.

Mirror skipped

The total number of bytes that have been skipped when performing a difference. These bytes are skipped because the data is not different on the source and target.

Replication status

- **Replicating**—Data is being replicated to the target.
- **Ready**—There is no data to replicate.
- **Pending**—Replication is pending.
- **Stopped**—Replication has been stopped.
- **Out of Memory—**Replication memory has been exhausted.
- **Failed**—The Double-Take service is not receiving replication operations from the Carbonite Availability driver. Check the Event Viewer for driver related issues.
- **Unknown—**The console cannot determine the status.

Replication queue

The total number of replication bytes in the source queue

Disk queue

The amount of disk space being used to queue data on the source

Bytes sent

The total number of mirror and replication bytes that have been transmitted to the target

Bytes sent compressed

The total number of compressed mirror and replication bytes that have been transmitted to the target. If compression is disabled, this statistic will be the same as **Bytes sent**.

Recovery point latency

The length of time replication is behind on the target compared to the source. This is the time period of replication data that would be lost if a failure were to occur at the current time. This value represents replication data only and does not include mirroring data. If you are mirroring and failover, the data on the target will be at least as far behind as the recovery point latency. It could potentially be further behind depending on the circumstances of the mirror. If mirroring is idle and you failover, the data will only be as far behind as the recovery point latency time.

Mirror start time

The UTC time when mirroring started

Mirror end time

The UTC time when mirroring ended

Total time for last mirror

The length of time it took to complete the last mirror process

Validating a full server to ESX job

Over time, you may want to confirm that any changes in your network or environment have not impacted your Carbonite Availability job. Use these instructions to validate an existing job.

1. From the **Jobs** page, highlight the job and click **View Job Details** in the toolbar.
2. In the **Tasks** area on the right on the **View Job Details** page, click **Validate job properties**.
3. Carbonite Availability validates that your source and target are compatible. The **Summary** page displays your options and validation items.

   Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

   Validation checks for an existing job are logged to the job log on the target server.
4. Once your servers have passed validation, click **Close**.

Editing a full server to ESX job

Use these instructions to edit a full server to ESX appliance job.

1. From the **Jobs** page, highlight the job and click **Edit Job Properties** in the toolbar. (You will not be able to edit a job if you have removed the source of that job from your Carbonite Replication Console session or if you only have Carbonite Availability monitor security access.)
2. You will see the same options for your full server to ESX job as when you created the job, but you will not be able to edit all of them. If desired, edit those options that are configurable for an existing job. See *Creating a full server to ESX job* on page 162 for details on each job option.

   Changing some options may require Carbonite Availability to automatically disconnect, reconnect, and remirror the job.

3. If you want to modify the workload items or replication rules for the job, click **Edit workload or replication rules**. Modify the **Workload item** you are protecting, if desired. Additionally, you can modify the specific **Replication Rules** for your job.

   Volumes and folders with a green highlight are included completely. Volumes and folders highlighted in light yellow are included partially, with individual files or folders included. If there is no highlight, no part of the volume or folder is included. To modify the items selected, highlight a volume, folder, or file and click **Add Rule**. Specify if you want to **Include** or **Exclude** the item. Also, specify if you want the rule to be recursive, which indicates the rule should automatically be applied to the subdirectories of the specified path. If you do not select **Recursive**, the rule will not be applied to subdirectories.

   You can also enter wildcard rules, however you should do so carefully. Rules are applied to files that are closest in the directory tree to them. If you have rules that include multiple folders, an exclusion rule with a wild card will need to be added for each folder that it needs applied to. For example, if you want to exclude all .log files from /home and your rules include /home, /home/folder1, and /home/folder2, you would need to add the exclusion rule for the root and each subfolder rule. So you will need to add exclude rules for /home/*.log , /home/folder1/*.log, and /home/folder2/*.log.

   If you need to remove a rule, highlight it in the list at the bottom and click **Remove Rule**. Be careful when removing rules. Carbonite Availability may create multiple rules when you are adding directories. For example, if you add /home/admin to be included in protection, then /home will be excluded. If you remove the /home exclusion rule, then the /home/admin rule will be removed also.

   Click **OK** to return to the **Edit Job Properties** page.

   If you remove data from your workload and that data has already been sent to the target, you will need to manually remove that data from the target. Because the data you removed is no longer included in the replication rules, Carbonite Availability orphan file detection cannot remove the data for you. Therefore, you have to remove it manually.

4. Click **Next** to continue.
5. Carbonite Availability validates that your source and target are compatible. The **Summary** page

displays your options and validation items.

Errors are designated by a white X inside a red circle. Warnings are designated by a black exclamation point (!) inside a yellow triangle. A successful validation is designated by a white checkmark inside a green circle. You can sort the list by the icon to see errors, warnings, or successful validations together. Click on any of the validation items to see details. You must correct any errors before you can continue. Depending on the error, you may be able to click **Fix** or **Fix All** and let Carbonite Availability correct the problem for you. For those errors that Carbonite Availability cannot correct automatically, you will need to modify the source or target to correct the error, or you can select a different target. You must revalidate the selected servers, by clicking **Recheck**, until the validation check passes without errors.

If you receive a path transformation error during job validation indicating a volume does not exist on the target server, even though there is no corresponding data being protected on the source, you will need to manually modify your replication rules. Go back to the **Choose Data** page and under the **Replication Rules**, locate the volume from the error message. Remove any rules associated with that volume. Complete the rest of the workflow and the validation should pass.

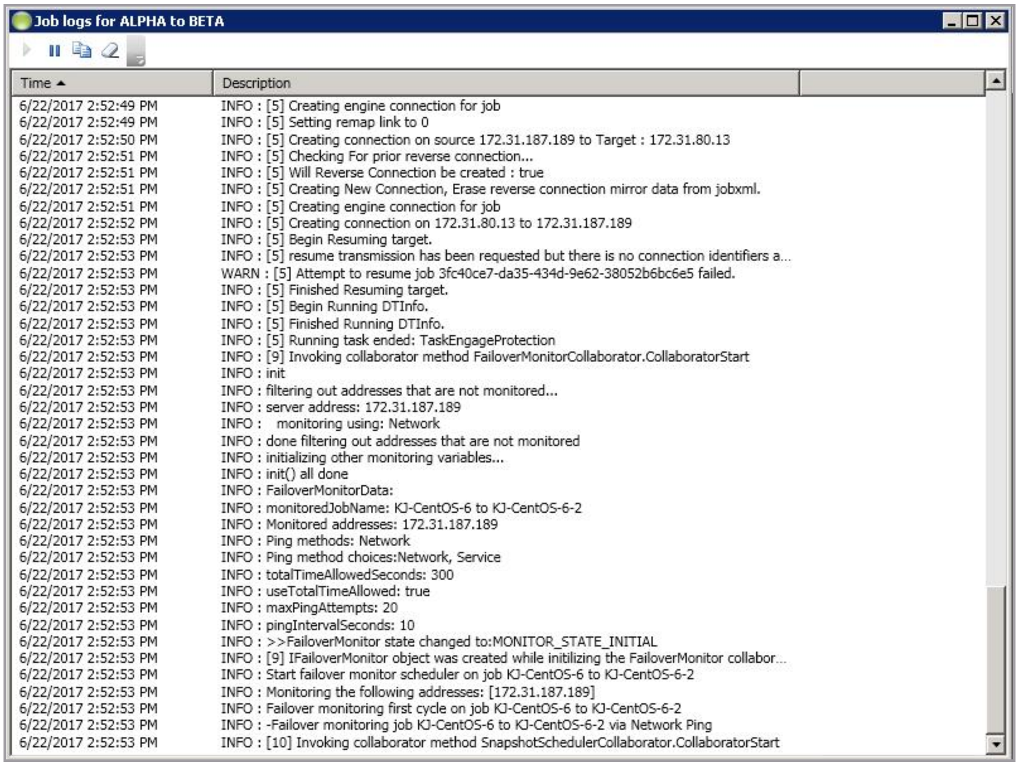
After a job is created, the results of the validation checks are logged to the job log. See the *Carbonite Availability and Carbonite Migrate Reference Guide* for details on the various Carbonite Availability log files.

6. Once your servers have passed validation and you are ready to update your job, click **Finish**.

Viewing a full server to ESX job log

You can view a job log file through the Carbonite Replication Console by selecting **View Job Log** from the toolbar on the **Jobs** page. Separate logging windows allow you to continue working in the Carbonite Replication Console while monitoring log messages. You can open multiple logging windows for multiple jobs. When the Carbonite Replication Console is closed, all logging windows will automatically close.

Because the job log window communicates with the target server, if the console loses communication with the target server after the job log window has already been opened, the job log window will display an error.



```
Time ▲                    Description
6/22/2017 2:52:49 PM      INFO : [5] Creating engine connection for job
6/22/2017 2:52:49 PM      INFO : [5] Setting remap link to 0
6/22/2017 2:52:50 PM      INFO : [5] Creating connection on source 172.31.187.189 to Target : 172.31.80.13
6/22/2017 2:52:51 PM      INFO : [5] Checking For prior reverse connection...
6/22/2017 2:52:51 PM      INFO : [5] Will Reverse Connection be created : true
6/22/2017 2:52:51 PM      INFO : [5] Creating New Connection, Erase reverse connection mirror data from jobxml.
6/22/2017 2:52:51 PM      INFO : [5] Creating engine connection for job
6/22/2017 2:52:52 PM      INFO : [5] Creating connection on 172.31.80.13 to 172.31.187.189
6/22/2017 2:52:53 PM      INFO : [5] Begin Resuming target.
6/22/2017 2:52:53 PM      INFO : [5] resume transmission has been requested but there is no connection identifiers a...
6/22/2017 2:52:53 PM      WARN : [5] Attempt to resume job 3fc40ce7-da35-434d-9e62-38052b6bc6e5 failed.
6/22/2017 2:52:53 PM      INFO : [5] Finished Resuming target.
6/22/2017 2:52:53 PM      INFO : [5] Begin Running DTInfo.
6/22/2017 2:52:53 PM      INFO : [5] Finished Running DTInfo.
6/22/2017 2:52:53 PM      INFO : [5] Running task ended: TaskEngageProtection
6/22/2017 2:52:53 PM      INFO : [9] Invoking collaborator method FailoverMonitorCollaborator.CollaboratorStart
6/22/2017 2:52:53 PM      INFO : init
6/22/2017 2:52:53 PM      INFO : filtering out addresses that are not monitored...
6/22/2017 2:52:53 PM      INFO : server address: 172.31.187.189
6/22/2017 2:52:53 PM      INFO :   monitoring using: Network
6/22/2017 2:52:53 PM      INFO : done filtering out addresses that are not monitored
6/22/2017 2:52:53 PM      INFO : initializing other monitoring variables...
6/22/2017 2:52:53 PM      INFO : init() all done
6/22/2017 2:52:53 PM      INFO : FailoverMonitorData:
6/22/2017 2:52:53 PM      INFO : monitoredJobName: KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM      INFO : Monitored addresses: 172.31.187.189
6/22/2017 2:52:53 PM      INFO : Ping methods: Network
6/22/2017 2:52:53 PM      INFO : Ping method choices:Network, Service
6/22/2017 2:52:53 PM      INFO : totalTimeAllowedSeconds: 300
6/22/2017 2:52:53 PM      INFO : useTotalTimeAllowed: true
6/22/2017 2:52:53 PM      INFO : maxPingAttempts: 20
6/22/2017 2:52:53 PM      INFO : pingIntervalSeconds: 10
6/22/2017 2:52:53 PM      INFO : >>FailoverMonitor state changed to:MONITOR_STATE_INITIAL
6/22/2017 2:52:53 PM      INFO : [9] IFailoverMonitor object was created while initilizing the FailoverMonitor collabor...
6/22/2017 2:52:53 PM      INFO : Start failover monitor scheduler on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM      INFO : Monitoring the following addresses: [172.31.187.189]
6/22/2017 2:52:53 PM      INFO : Failover monitoring first cycle on job KJ-CentOS-6 to KJ-CentOS-6-2
6/22/2017 2:52:53 PM      INFO : -Failover monitoring job KJ-CentOS-6 to KJ-CentOS-6-2 via Network Ping
6/22/2017 2:52:53 PM      INFO : [10] Invoking collaborator method SnapshotSchedulerCollaborator.CollaboratorStart
```

The following table identifies the controls and the table columns in the **Job logs** window.

Start ▶

This button starts the addition and scrolling of new messages in the window.

Pause ⏸

This button pauses the addition and scrolling of new messages in the window. This is only for the **Job logs** window. The messages are still logged to their respective files on the server.

Copy

This button copies the messages selected in the **Job logs** window to the Windows clipboard.

Clear

This button clears the **Job logs** window. The messages are not cleared from the respective files on the server. If you want to view all of the messages again, close and reopen the **Job logs** window.

Time

This column in the table indicates the date and time when the message was logged.

Description

This column in the table displays the actual message that was logged.

Failing over full server to ESX jobs

When a failover condition has been met, failover will be triggered automatically if you disabled the wait for user option during your failover configuration. If the wait for user before failover option is enabled, you will be notified in the console when a failover condition has been met. At that time, you will need to trigger it manually from the console when you are ready.

If you have paused your target, failover will not start if configured for automatic failover, and it cannot be initiated if configured for manual intervention. You must resume the target before failover will automatically start or before you can manually start it.

1. On the **Jobs** page, highlight the job that you want to failover and click **Failover or Cutover** in the toolbar.
2. Select the type of failover to perform.
    - **Failover to live data**—Select this option to initiate a full, live failover using the current data on the target. This option will shutdown the source machine (if it is online), stop the protection job, and start the replica virtual machine on the target with full network connectivity.
    - **Perform test failover**—Select this option to perform a test failover.
        - All data volumes must be under LVM for test failover.
        - In order to use test failover, you must make sure your target has sufficient free disk space available. The free space on each volume group on the target must be larger than 50% of the total size of all logical volumes in that volume group on the source. Meeting that amount of free space may depend on the **Disk Configuration Strategy** you selected under **Replica Virtual Disk Volumes**.
            - **Create disks matching source**—With the match source option, you must have sufficient free space on the source before the job is created because the target disks will be matching the source. You may need to increase free disk space on the source (perhaps add a partition which has been created on the raw disk on the source and then extend the volume group), in order to allow for sufficient free space to be matched on the target.
            - **Create disks per volume**—With the per volume option, select a volume group size on the target that has enough free space to accommodate the 50% free space requirement.
        - Test failover is not supported for Btrfs file systems.
        - The source, target, and protection job will remain online and uninterrupted during the test.
        - The test will be performed using the test failover settings configured during job creation.
        - The test will use the current data on the target.
        - Scheduled snapshots will be deferred during the test and taken automatically after the test is undone.

    - The test failover will take a snapshot of the current data on the target and create a new set of virtual disks. The data from the snapshot will be mirrored to the new set of disks using the same mirroring options as the protection job.
    - Once the mirror is complete, the replica virtual machine or alternate replica virtual machine, depending on your selected configuration, is automatically brought online using the new set of disks.
    - The replica virtual machine or alternate replica virtual machine, depending on your selected configuration, will use the network settings specified in the test failover settings of the protection job.
    - When you are finished with your test, undo it.
    - When you undo a test failover, the new set of disks will be maintained or deleted as specified in the test failover settings of the protection job.
    - At any time during a test failover, you can undo the test, perform a live failover, or failover to a snapshot. (Performing a live failover or failing over to a snapshot will automatically undo any test in progress.)
  - **Failover to a snapshot**—Select this option to initiate a full, live failover without using the current data on the target. Instead, select a snapshot and the data on the target will be reverted to that snapshot. This option will not be available if there are no snapshots on the target. To help you understand what snapshots are available, the **Type** indicates the kind of snapshot.
    - **Scheduled**—This snapshot was taken as part of a periodic snapshot.
    - **Deferred**—This snapshot was taken as part of a periodic snapshot, although it did not occur at the specified interval because the job between the source and target was not in a good state.
    - **Manual**—This snapshot was taken manually by a user.

3. Select how you want to handle the data in the target queue.
    - **Apply data in target queues before failover or cutover**—All of the data in the target queue will be applied before failover begins. The advantage to this option is that all of the data that the target has received will be applied before failover begins. The disadvantage to this option is depending on the amount of data in queue, the amount of time to apply all of the data could be lengthy.
    - **Discard data in the target queues and failover or cutover immediately**—All of the data in the target queue will be discarded and failover will begin immediately. The advantage to this option is that failover will occur immediately. The disadvantage is that any data in the target queue will be lost.
4. When you are ready to begin

Once failover has started, do not reboot the target appliance. If the failover process is interrupted, it may fail.

---

5. If you performed a test failover, you can undo it by selecting **Undo Failover or Cutover** in the toolbar. The replica virtual machine on the target will be shut down and deleted if you chose to use an alternate virtual machine for the test. The virtual disks for used for the test will be deleted.

If you need to update DNS after failover, there is a sample DNS update script located in /etc/DT/sysprep.d. You may need to modify the script for your environment. If you need basic assistance with script modifications, contact technical support. Assistance with advanced scripting will be referred to Professional Services.

There is no reverse or failback once you have failed over.

Reversing protection after failover for full server to ESX jobs

There is no automated reverse or failback for a full server to ESX appliance job once you have failed over. If you need to go back to your original hardware, you will need to create a new job in the opposite direction following one of the processes below, depending on the original source.

- **Physical server**—Use these steps if your original source is a physical server.
    1. Resolve the problems on the original source that caused it to fail. If you need to deploy a new server, use the same operating system and disk configuration as the original source.
    2. If Carbonite Availability is still running on the original source, replace the license since that license is currently running on the failed over server. If Carbonite Availability is no longer installed, reinstall it with an appropriate license.
    3. Create a full server job from the failed over server to your original source. See *Creating a full server job* on page 114 for details on creating this job.
    4. Once the initial mirror is complete, failover the full server job. See *Failing over full server jobs* on page 150 for details on this process.
- **ESX virtual server**—Use these steps if your original source is a virtual server on an ESX host.
    1. Delete the original source virtual server from the ESX host. If you want to reuse the .vmdk files again, only delete the original source virtual server from the ESX inventory.
    2. If you do not have one already, create a virtual recovery appliance on the ESX host where the original source virtual server is located. This appliance will be the target of the new job you are going to create. This appliance needs Carbonite Availability installed and licensed on it. For more details, see *Full server to ESX requirements* on page 154.
    3. Create a full server to ESX job from your failed over server to the appliance on the ESX host where the original source virtual server is located. See *Creating a full server to ESX job* on page 162 for details on creating this job.
    4. Once the initial mirror is complete, failover the full server to ESX job. See *Failing over full server to ESX jobs* on page 206 for details on this process.
- **Hyper-V virtual server**—Use these steps if your original source is a virtual server on a Hyper-V host.
    1. Resolve the problems on the original source that caused it to fail. If you need to deploy a new server, use the same operating system and disk configuration as the original source.
    2. If Carbonite Availability is still running on the original source, replace the license since that license is currently running on the failed over server. If Carbonite Availability is no longer installed, reinstall it with an appropriate license.
    3. Create a full server job from the failed over server to your original source. See *Creating a full server job* on page 114 for details on creating this job.
    4. Once the initial mirror is complete, failover the full server job. See *Failing over full server jobs* on page 150 for details on this process.

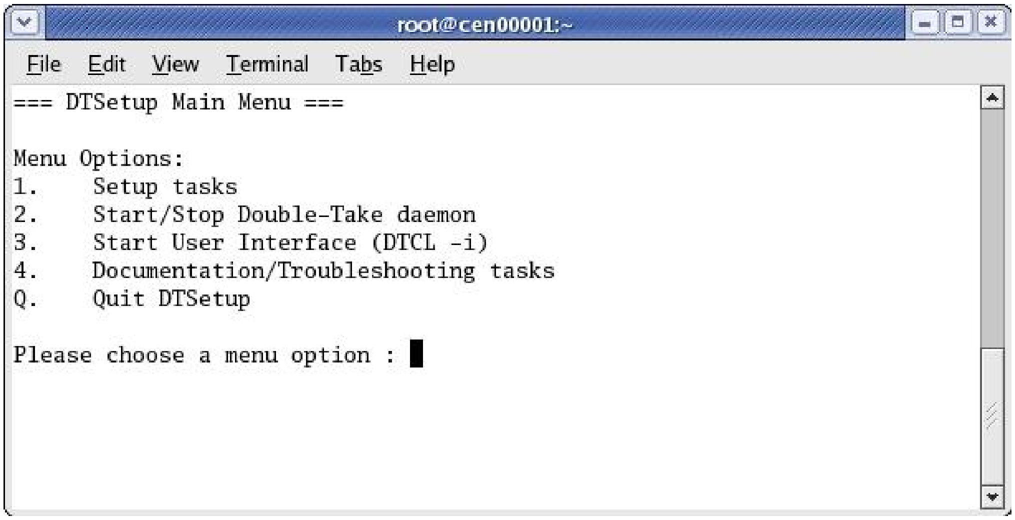
Chapter 7 DTSetup

DTSetup is a menu-driven application that provides easy access to Carbonite Availability server configuration. Select a link for more information on DTSetup configuration tasks.

- *Running DTSetup* on page 211—This topic includes instructions for launching DTSetup.
- *Setup tasks* on page 212—The setup tasks allow you to configure license keys, security groups, block device replication configuration, server configuration, and driver performance settings.
- *Starting and stopping the service* on page 217—Built-in scripts allow you to quickly and easily start and stop the Carbonite Availability service.
- *Starting DTCL* on page 218—You can launch the Carbonite Availability interactive command prompt which allows you to enter DTCL commands one at a time.
- *Viewing documentation and troubleshooting tools* on page 219—DTSetup provides easy access to Carbonite Availability log files, a diagnostic collection tool, and several legal documents.
- *DTSetup menus* on page 220—This topic includes a list overview of the DTSetup menu system. Reference the links in the list for complete details on completing tasks in DTSetup.

Running DTSetup

1. Run the DTSetup command from the shell prompt to start DTSetup. The command is case-sensitive.
2. The first time you run DTSetup after an installation, you will be prompted to review the Carbonite license agreement. Review the agreement and accept the terms of agreement by typing yes. You cannot use Carbonite Availability without agreeing to the licensing terms.
3. When the DTSetup menu appears, enter the number of the menu option you want to access.



```
=== DTSetup Main Menu ===

Menu Options:
1.    Setup tasks
2.    Start/Stop Double-Take daemon
3.    Start User Interface (DTCL -i)
4.    Documentation/Troubleshooting tasks
Q.    Quit DTSetup

Please choose a menu option :
```

Setup tasks

The setup tasks are generally configured once. Select a link below to learn more about that setup task.

- *Activating your server* on page 213—License keys and activation keys license and activate your Carbonite Availability servers.
- *Modifying security groups* on page 214—Security groups provide access to Carbonite Availability.
- *Configuring server settings* on page 215—If desired, you can modify server settings through the Carbonite Availability configuration file.
- *Configuring driver performance settings* on page 216—If desired, you can specify Carbonite Availability driver performance settings.

Activating your server

Before you can use Carbonite Availability, each source and target server must have a valid license key, which is an alpha-numeric codes that applies the appropriate Carbonite Availability license to your installation.

1. Start DTSetup. See *Running DTSetup* on page 211.
2. Select **Setup tasks**.
3. Select **Set License Key Menu**.
4. Select **Set License Key in /etc/DT/DT.conf**.
5. Enter your license key and press Enter. The license key will automatically be inserted into the configuration file. You are prompted to start the Carbonite Availability service after the first installation, and you must restart the service each time the license key is modified, such as after an upgrade.
6. Press Enter to return to the menu.
7. Press Q as many times as needed to return back to the main menu or to exit DTSetup.

Modifying security groups

During the installation, the user root is automatically added to the Carbonite Availability administrators security group. If you want to add other users or remove root, you will need to modify the security group configuration for each source and target server. See *Security* on page 221 for more details on the security groups and the privileges granted to each group.

1. Start DTSetup. See *Running DTSetup* on page 211.
2. Select **Setup tasks**.
3. Select **Add/Remove users to Double-Take groups**.
4. Select the necessary menu options to add or remove groups to the administrator or monitors group as needed, and specify the user name when prompted.
5. When you have completed your security group modifications, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Configuring server settings

Server settings are available in various places. You can access them via the Replication Console for Linux, through DTCL, or through DTSetup. Initially, the server settings file, /etc/DT/DT.conf, on the source and target is blank. To populate it with default values, start and stop the Double-Take service once.

1. Start DTSetup. See *Running DTSetup* on page 211.
2. Select **Setup tasks**.
3. Select **Edit Double-Take config file**.
4. The server settings are listed in alphabetical order. Make modifications as necessary, using the control keys specified at the bottom of the page. For a complete list of each server setting, valid values, default values, and optional notes, see *Server Settings* in the *Scripting Guide*.
5. Press control-X to exit the configuration file.
6. Enter Yes or No to save any changes.
7. Press Q as many times as needed to return back to the main menu or to exit DTSetup.

Configuring driver performance settings

Driver settings provide configuration flexibility so you can adjust Carbonite Availability based on your servers, network, and replication requirements. You may want to modify driver settings on both the source and target.

> Changing the driver performance settings can have a positive or negative impact on server performance. These settings are for advanced users. If you are uncertain how to best modify the driver performance settings, contact technical support.

1. Start DTSetup. See *Running DTSetup* on page 211.
2. Select **Setup tasks**.
3. Select **Configure Double-Take driver performance**.
4. The current driver settings are displayed.
5. Select a driver setting to modify the option.
    - **Toggle Adaptive Throttling**—You can toggle between enabling (true) and disabling (false) **Adaptive Throttling**. This occurs when kernel memory usage exceeds the **Throttling Start Level** percentage. When throttling is enabled, operations are delayed by, at most, the amount of time set in **Maximum Throttling Delay**, thus reducing kernel memory usage. Throttling stops when the kernel memory usage drops below the **Throttling Stop Level** percentage.
    - **Toggle Forced Adaptive Throttling**—You can toggle between enabling (true) and disabling (false) **Forced Adaptive Throttling**. This causes all operations to be delayed by, at most, the amount of time in set in **Maximum Throttling Delay**, regardless of the kernel memory being used. **Adaptive Throttling** must be enabled (true) in order for **Forced Adaptive Throttling** to work.
    - **Set Maximum Throttling Delay**—This option is the maximum time delay, in milliseconds, used by the driver during a system delay.
    - **Set Throttling Delay Interval**—This option is the interval, in milliseconds, to check memory usage during a throttling delay. If a delay is no longer needed, the remainder of the delay is skipped.
    - **Set Throttling Start Level**—Throttling starts when disk writes reach the specified percentage. This prevents the driver from stopping replication because memory has been exhausted.
    - **Set Throttling Stop Level**—Throttling stops when disk writes reach the specified percentage.
    - **Set Memory Usage Limit**—This option is the amount of kernel memory, in bytes, used for queuing replication operations. When this limit is exceeded, the driver will send an error to the service forcing a remirror of all active connections.
    - **Set Maximum Write Buffer Size**—This option is the maximum amount of system memory, in bytes, allowed for a single write operation. Operations exceeding this amount are split into separate operations in the queue.
6. After you have completed your driver performance modifications, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Starting and stopping the service

The Double-Take service will start automatically after Carbonite Availability is installed and the server is rebooted. You can start and stop the Double-Take service using this built-in DTSetup script.

1. Start DTSetup. See *Running DTSetup* on page 211.
2. Select **Start/Stop Double-Take service**.
3. Select the necessary menu option to start or stop the service and handle the driver configuration.
    - **Start Double-Take and process driver config**—This option starts the Double-Take service and loads the Carbonite Availability drivers.
    - **Stop Double-Take but preserve driver config**—This option stops the Double-Take service but does not unload the Carbonite Availability drivers.
    - **Restart service but preserve driver config**—This option does a full stop and start of the Double-Take service but does not unload the Carbonite Availability drivers.
    - **Restart service and reset driver config—**This option does a full stop and start, completely unloading the Double-Take service and Carbonite Availability drivers and then reloading them.
    - **Stop the running service and teardown driver config**—This option stops the Double-Take service and the Carbonite Availability drivers are unloaded.
    - **Go to Replication Configuration menu**—This option takes you to **Setup Tasks**, **Configure Block Device Replication**. When you press Q to exit from that menu, you will return this menu.
4. When you have completed your starting and stopping tasks, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Starting DTCL

You can launch the Carbonite Availability interactive command prompt which allows you to enter DTCL commands one at a time.

1. Start DTSetup. See *Running DTSetup* on page 211.
2. Select **Start User Interface (DTCL -i)**.
3. Enter your DTCL commands one at a time at the **Command** prompt. For a complete list of DTCL commands, their syntax, and instructions for completing tasks using DTCL, see the *Scripting Guide*.
4. To exit the DTCL **Command** prompt, type exit.
5. When you have completed your DTCL tasks, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Viewing documentation and troubleshooting tools

1. Start DTSetup. See *Running DTSetup* on page 211.
2. Select **Documentation/Troubleshooting tasks**.
3. Select **View log files** to view the following log files. Carbonite Availability logs alerts, which are processing notifications, warnings, and error messages. The logs are written to disk.
    - **View /*.dtl in less**—This option uses the less file viewer program to view all of the Carbonite Availability logs, starting from the most recent.
    - **Follow the output of latest**—This option uses tail -f to watch the output of the Carbonite Availability logs in real-time.
    - **View /var/log/messages in less**—This option uses the less file viewer program to view the system log messages.
    - **Follow the output of /var/log/messages**—This option uses tail -f to watch the output of the system log messages in real-time.
4. Select one of the **Collect and package diagnostic info** selections to run the DTInfo script which collects configuration data. This can be useful when reporting problems to technical support. Depending on the diagnostic option you select, the amount of data to be collected varies between basic, detailed and full diagnostic information. You must have root (or uid 0 equivalent) to execute the diagnostics or to copy or read the resulting file.
5. Select **View user documentation** to view several legal documents. DTSetup attempts to determine your viewers, although you can specify your viewer.
    - **View End User License Agreement TXT**—This option views the End User License Agreement legal document.
    - **View driver module license TXT**—This option views the open source legal document.
    - **Change a document viewer**—This option allows you to specify a document viewer.
6. When you have completed your documentation and troubleshooting tasks, press Q as many times as needed to return back to the main menu or to exit DTSetup.

DTSetup menus

The following lists is an overview of the DTSetup menu system. Reference the links for complete details on completing tasks in DTSetup.

1. **Setup tasks—**License keys, security groups, replication configuration, server configuration, and driver performance settings. See *Setup tasks* on page 212.
    1. **Set License Key Menu**—See *Activating your server* on page 213.
    2. **Add/Remove users to Double-Take groups**—See *Modifying security groups* on page 214.
    3. **Edit Double-Take config file**—See *Configuring server settings* on page 215.
    4. **Configure Double-Take driver performance—**See *Configuring driver performance settings* on page 216.
2. **Start/Stop Double-Take service**—See *Starting and stopping the service* on page 217.
3. **Start User Interface (DTCL -i)**—See *Starting DTCL* on page 218.
4. **Documentation/Troubleshooting tasks—**See *Viewing documentation and troubleshooting tools* on page 219.

Chapter 8 Security

To ensure protection of your data, Carbonite Availability offers multi-level security using native operating system security features. Privileges are granted through membership in user groups defined on each machine. To gain access to a source or target, the user must provide a valid local user account that is a member of one of the Carbonite Availability security groups. Once a valid user name and password have been provided and the source or target has verified membership in one of the security groups, the user is granted appropriate access to the source or target and the corresponding features are enabled in the client. Access is granted on one of the following three levels.

- **Administrator Access**—All features are available for that machine.
- **Monitor Access**—Servers and statistics can be viewed, but functionality is not available.
- **No Access**—Servers appear in the clients, but no access to view the server details is available.

Although passwords are encrypted when they are stored, Carbonite security design does assume that any machine running the client application is protected from unauthorized access. If you are running the client and step away from your machine, you must protect your machine from unauthorized access.

Adding users to the security groups

The security groups are automatically created during the installation process. The groups can be local or LDAP (Lightweight Directory Access Protocol). The groups are called dtadmin (default group ID 501) and dtmon (default group ID 502). During the installation, the user root is automatically added to the dtadmin group.

Users that need administrator access to Carbonite Availability must be added to the dtadmin group. Users that need monitor only access must be added to the dtmon group. In both cases, you must provide a valid local user account.

1. Run the DTSetup command from the shell prompt. The command is case-sensitive.
2. Select **Setup tasks**.
3. Select **Add/Remove users to Double-Take groups**.
4. Select the necessary menu options to add or remove groups to the administrator or monitors group as needed, and specify the user name when prompted.
5. When you have completed your security group modifications, press Q as many times as needed to return back to the main menu or to exit DTSetup.

Chapter 9 Special network configurations

Carbonite Availability can be implemented with very little configuration necessary in small or simple networks, but additional configuration may be required in large or complex environments. Because an infinite number of network configurations and environments exist, it is difficult to identify all of the possible configurations. Review the following sections for configuration information for that particular type of network environment.

- *Firewalls* on page 224
- *NAT* on page 225

Firewalls

If your source and target are on opposite sides of a firewall, you will need to configure your hardware to accommodate communications. You must have the hardware already in place and know how to configure the hardware ports. If you do not, see the reference manual for your hardware.

- **Carbonite Availability ports**—Ports 1500, 1505, 1506, 6325, and 6326 are used for Carbonite Availability communications and must be open on your firewall. Open UDP and TCP for both inbound and outbound traffic.
- **ESX ports**—If you are using VirtualCenter or an ESX host, port 443 is also required and must be opened.

You need to configure your hardware so that the Carbonite Availability ports and ESX ports applicable to your environment are open. Since communication occurs bidirectionally, make sure you configure both incoming and outgoing traffic.

There are many types of hardware on the market, and each can be configured differently. See your hardware reference manual for instructions on setting up your particular router.

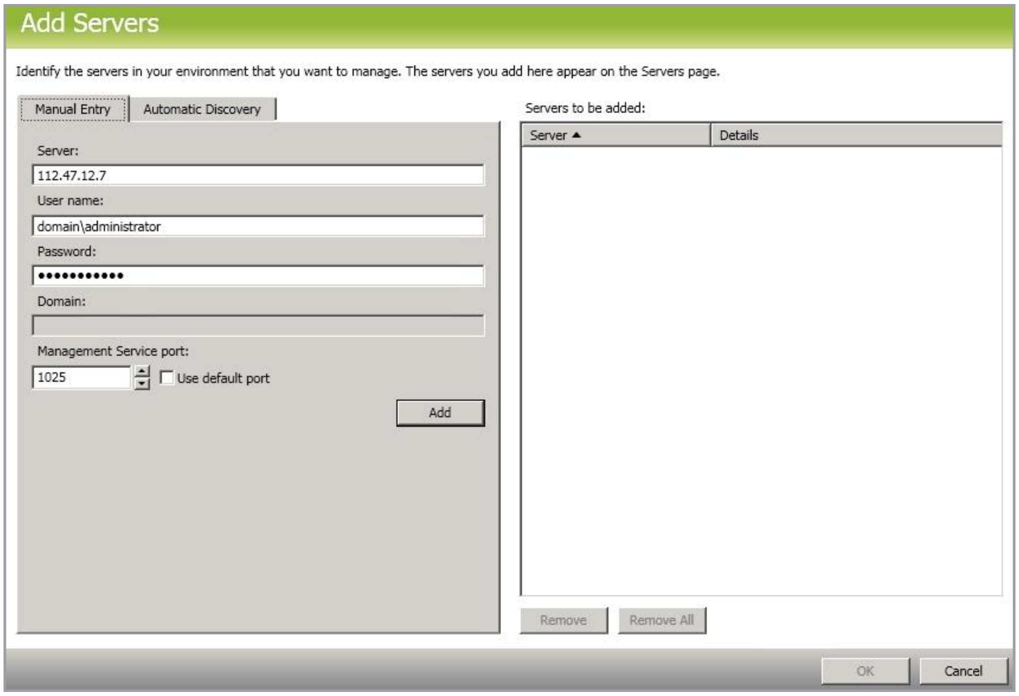
NAT

As outlined in the requirements, Carbonite Availability supports NAT environments with the following caveats.

- Only IPv4 is supported.
- Only standalone servers are supported.
- DNS failover and updates will depend on your configuration
    - Only the source or target can be behind a router, not both.
    - The DNS server must be routable from the target

When setting up a job in an environment with IP or port forwarding, make sure you specify the following configurations.

- Make sure you have added your server to the Carbonite Replication Console using the correct public or private IP address. The name or IP address you use to add a server to the console is dependent on where you are running the console. Specify the private IP address of any servers on the same side of the router as the console. Specify the public IP address of any servers on the other side of the router as the console. This option is on the **Add Servers** page in the **Manual Entry** tab.



Add Servers

Identify the servers in your environment that you want to manage. The servers you add here appear on the Servers page.

Server: 112.47.12.7

User name: domain\administrator

Password: •••••••••••

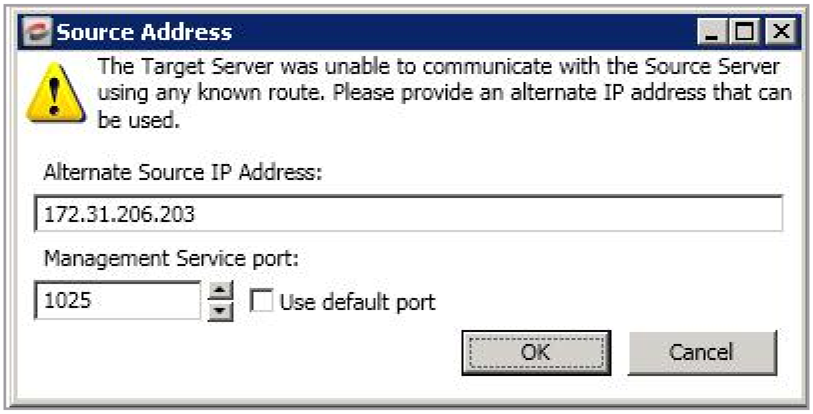
Domain:

Management Service port: 1025 □ Use default port

Servers to be added:

Server ▲	Details

- When choosing the target server for your job, you may be prompted for a route from the target to the source. This route, and a port if you are using a non-default port, is used so the target can communicate with the source to build job options. This dialog box will be displayed, only if needed, after you click **Next** on the **Choose Target** page in the job creation wizard.



The Target Server was unable to communicate with the Source Server using any known route. Please provide an alternate IP address that can be used.

Alternate Source IP Address:
172.31.206.203

Management Service port:
1025
□ Use default port