



# Bacula Enterprise Fundamentals

**Bacula Systems Documentation**

---

# Contents

<b>1</b>	<b>About Bacula Enterprise</b>	<b>2</b>
<b>2</b>	<b>Bacula Enterprise Architecture</b>	<b>11</b>
<b>3</b>	<b>Bacula Enterprise Explained</b>	<b>17</b>

# Contents

---

The following chapter aims at presenting you with all the necessary information to start with Bacula Enterprise solution. To gain full understanding of Bacula, it is recommended to follow the order of browsing through the articles presented below.

1. Go through basic information in the article *About Bacula Enterprise*.
2. Familiarize yourself with Bacula Enterprise architecture in the *Bacula Enterprise Architecture*.
3. Learn about how Bacula Enterprise works in the article *Bacula Enterprise Explained*.

After you got acquainted with the Fundamentals chapter, you may want to read more detailed documentation.

## 1 About Bacula Enterprise

The following document aims at introducing the reader to what Bacula Enterprise, or simply: Bacula -is, and what its main features are. It is meant for anybody who wants to have basic understanding of Bacula.

### 1.1 What is Bacula Enterprise?

Bacula is a highly secure, enterprise-class, feature-rich software that allows system administrators to easily manage backup, recovery, and verification of computer data across a network of computers of different kinds. Bacula can run entirely upon a single computer and can backup to various types of media, including tape and disk. It is independent of both hardware and infrastructure, can run on any brand of computer and fits neatly into on-premise as well as hybrid deployment or full cloud infrastructure. It is especially stable, reliable, secure and massively scalable. The broad support of plugins enormously simplifies the backup of complex sites and technologies. Bacula is used in both small deployments and in extremely large organizations, as well as HPC (High Performance Computing) environments.

### 1.2 Features

Bacula brings rich new features and enhancements for backup and data recovery to enterprises, data centers and managed service providers. A comprehensive list of features, organized by categories is presented below:

## Core Features

The following article presents the core features of Bacula.

Bacula is:

- Especially secure – Bacula’s modular architecture is designed to offer higher levels of security than other solutions. It also has a broad set of additional and specific security features to enable IT department leaders to implement a high security strategy that meets best practices and zero-trust policies.
- Network-based – all backups and restores are done via the network. This allows Bacula to run on a single server, and backup any computer in your data center.
- Centrally administered – Bacula administration is centralized in the Director. Centralizing the administration is essential when your network grows beyond a few computers.
- Run automatically – once set up, Bacula runs automatically. Normally, a properly configured Bacula installation requires little maintenance or intervention, except when adding new machines, or when hardware errors occur.
- Able to perform bookkeeping – Bacula does hard bookkeeping by maintaining a catalog of what is backed up and where. If you have hundreds or thousands of machines to be backed up, it is essential that the bookkeeping is automatically maintained by a program rather than requiring human intervention.
- Responsive to multiple platforms - Bacula is compatible with a wide range of platforms: BSD, Unix, Linux, MS Windows, Mac OS X, and others as well as a large range of hardware.
- Modularly designed – means it scales well from small shops (one machine) to very large ones (tens of thousands of machines).
- Responsive to multiple backup media – Bacula handles a variety of different media such as local disks, network disks, block storage, tapes, autochangers or cloud devices.
- Reliable – Bacula consists of advance tools for memory and lock management. It is very common to run it without problems for months.
- Characterized by high performance – Bacula’s modern multi-threaded design allows running multiple simultaneous backups and can achieve high speeds writing to disk, tape or cloud.
- Customizable – Bacula can be easily customized to almost any backup/restore need.
- Able to restore data rapidly - easy and fast restores using Bacula’s database and graphical user interface.
- Equipped with advanced reporting, notification, monitoring systems – Bacula has excellent reporting addons such as Bweb, which allows to get graphical and raw statistics for custom reports, billings, trends, optimizations and capacity planning. Bacula provides very good notifications and monitoring capabilities, and can also be extremely well-integrated into monitoring tools such as Nagios.
- Providing extremely good interoperability through:
  - Full featured REST API
  - Full featured Console User Interface
  - Script-friendly configuration files
  - Ability to expand any job capabilities with additional pre or post script
- Including advanced backup policies:
  - Progressive Virtual Full backup syndication
  - Differential, Full, Incremental backup types
  - Built-in Job scheduler
    - \* Job multiplexing (run hundreds of jobs simultaneously)

- Re-scheduling capabilities
- Advanced and customizable features to control job concurrency
- Including advanced operation capabilities, such as:
  - Restart incomplete jobs features
  - Bandwidth limitation
  - Job sequencing using priorities
  - Customize restore operations to the original source or to a different target
- Enormously scalable - Bacula is scalable from small single computer systems to systems consisting of hundreds or thousands of computers located over a large network.
- Its highly modular architecture is specifically designed to manage:
  - Billions of file/object records in the catalog
  - Large amounts of data (PB per storage)
  - Thousands of Clients per Director
  - Each Daemon/Service can be separated on the network
  - Ability to extract single files from snapshots such as virtual machine

**See also:**

Go to:

- *Security Features*
- *Storage Destinations*
- *NAS and External Protection*
- *Endpoint Features*
- *Cloud and Virtualization*
- *Databases*
- *Supported Applications*

Go back to the *About Bacula Enterprise* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## Security Features

The following article presents the security features of Bacula.

Bacula is a solution known by its especially strong security standards that allow not only to keep data safe, but also all the infrastructure around it. Because of its modern and advanced security features, Bacula is relied on by many of the largest government and defence organizations. Some more details about security related features are presented below:

- Distributed and isolated *architecture* with limited privileges among each component
- FIPS 140-2 compliance
- Automatic encryption for all network communications (can be turned off or modified with custom certificates)
- Verification of files previously catalogued:
  - File integrity purposes to detect silent data corruption

- System break-in detection (Tripwire-like capability)
- CRAM-MD5 password authentication between each component (daemon)
- Configurable Data encryption at rest on a Client by Client basis
- Configurable Data encryption at rest globally at Storage Daemon level
- Computation of MD5, SHA1, SHA256 or SHA512 signatures of the file data
- Immutable disk volume feature for Linux based storage destinations
- *Immutable NAS Support* (Netapp SnapLock, DataDomain RetentionLock or HPE StoreOnce Catalyst among others)
- Immutable tape support (WORM)
- Immutable cloud support (S3 ObjectLock, Azure Blob immutable)
- Connectivity to Active Directory or LDAP services to protect access
- 2-Factor authentication through One-Time Password (OTP), allowing use of smartphones with bio-metric functions to access Bacula's web GUI
- Advanced Ransomware protection tools, such as:
  - Security module to detect vulnerabilities, bad configurations, missing updates and many other threats on Windows and Linux
  - Automatic malware protection by known hash checking for backup, restore and verify processes
  - Antivirus Plugin to detect malware on stored data
- Advanced File Daemon restriction mechanisms to limit backup, restore or scripts scope by path, user or group id
- Monitoring integrations with SNMP and Security Information and Event Management (SIEM) Systems
- Agnostic N-Tier backup support including offsite and/or cloud copies.
- Backup poisoning detection (described here)
- Automatic security configuration assessment (described here)
- Console Directory authentication (ldap/ad director connector through Bacula Pluggable Authentication Module).
- Auditory logging (through Event Messages)
- User Role Base Access capabilities through resource ACLs

**See also:**

Go back to:

- *Core Features*

Go to:

- *Storage Destinations*
- *NAS and External Protection*
- *Endpoint Features*
- *Cloud and Virtualization*
- *Databases*
- *Supported Applications*

Go back to the *About Bacula Enterprise* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## Storage Destinations

The following article presents information on Bacula storage destinations.

- Global Endpoint Deduplication™ (deduplication from the client to the storage)
  - Bacula Enterprise native deduplication system, integrated into the core of the solution
  - Deduplication across any kind of datasource, globally to a Storage Daemon
  - Deduplication in the source: just send changed data through the network
  - VirtualFull or Synthetic backup integration: just copy references for extremely fast jobs
  - Ready to be used to dedup in the cloud
- Data compression
- Communication line compression, allowing users to divide by 3 the volume of data transmitted across communication lines
- Filesystems deduplication (Aligned Volume Format)
- Integration with practically any kind of tape storage
- Storage Daemon to Storage Daemon (SD2SD) for replication capabilities, using deduplication
- Storage Daemon can be deployed in any Linux systems
- Storage Daemon Reporting capabilities to retrieve a range of information such as available disk space and disk usage information
- SAN Shared Tape Library Module
- Oracle ACSLS support
- Full Hybrid Cloud capability, via S3, Azure, Glacier, Google Cloud and Oracle Cloud interfaces.

### See also:

Go back to:

- *Core Features*
- *Security Features*

Go to:

- *NAS and External Protection*
- *Endpoint Features*
- *Cloud and Virtualization*
- *Databases*
- *Supported Applications*

Go back to the *About Bacula Enterprise* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## NAS and External Protection

The following article presents information on NAS and external protection.

Bacula can also use NAS and other storage systems as the source of information and protect them using specific techniques. Among them:

- Nutanix Filer
- Netapp Filer
- NDMP support
- NDMP CAB support
- Hadoop HDFS module
- Incremental Accelerator for NetApp
- S3 Objects Backup (from Amazon or any generic solution)
- OpenStack Swift Backup

### See also:

Go back to:

- *Core Features*
- *Security Features*
- *Storage Destinations*

Go to:

- *Endpoint Features*
- *Cloud and Virtualization*
- *Databases*
- *Supported Applications*

Go back to the *About Bacula Enterprise* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## Endpoint Features

The following article presents the endpoint features of Bacula.

Some relevant features about backup and restoring endpoint servers:

- Continuous Data Protection
- VSS Support in Windows systems
- Client behind NAT support
- Flexibility to configure who initiates the connection (Storage calls client)
- BareMetal Recovery for Linux hosts
- BareMetal Recovery for Windows hosts

### See also:

Go back to:

- *Core Features*
- *Security Features*
- *Storage Destinations*
- *NAS and External Protection*

Go to:

- *Cloud and Virtualization*
- *Databases*
- *Supported Applications*

Go back to the *About Bacula Enterprise* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## Cloud and Virtualization

The following article presents information on Bacula cloud and virtualization.

Most of the systems in the IT world can be backed up following an agent based strategy, where the File Daemon is deployed in the target host, independently from the underlying platform.

However, Bacula also offers a very complete solution to protect Virtual Machines or Containers in agent-less mode and in a consistent way through the usage of snapshot techniques. These technologies are covered through specific Plugins. The main Virtualization platforms in the market are covered:

- Hypervisors
  - VMware
  - Hyper-V
  - Nutanix AHV
  - Xen
  - Proxmox
  - KVM
  - Red Hat Virtualization/oVirt
- Cloud Infrastructure as a Service
  - Azure VM
- Container Environments
  - Docker
  - Kubernetes Clusters
  - Red Hat OpenShift

**See also:**

Go back to:

- *Core Features*
- *Security Features*
- *Storage Destinations*



- *NAS and External Protection*
- *Endpoint Features*

Go to:

- *Databases*
- *Supported Applications*

Go back to the *About Bacula Enterprise* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## Databases

The following article presents information on databases.

Databases have specific requirements in order to be backed up or restored. Bacula uses corresponding specific techniques to provide those backup and restore capabilities through specific plugins. The main Databases in the market are covered:

- MSSQL
- PostgreSQL
- MySQL
- Oracle
- SAP HANA
- SAP (Sybase) ASE
- MySQL Percona
- Maria DB
- IBM DB2

### See also:

Go back to:

- *Core Features*
- *Security Features*
- *Storage Destinations*
- *NAS and External Protection*
- *Endpoint Features*
- *Cloud and Virtualization*

Go to:

- *Supported Applications*

Go back to the *About Bacula Enterprise* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## Supported Applications

The following article presents information on supported applications.

Bacula provides also specific plugins to handle the backup and restore of specific applications on-premise or in the cloud (such as Software as a Service). The following list of applications is supported:

- SaaS
  - Microsoft 365
  - Google Workspace
- On-premise
  - Microsoft Exchange
  - Microsoft Sharepoint
  - Microsoft Active Directory
  - LDAP

### See also:

Go back to:

- *Core Features*
- *Security Features*
- *Storage Destinations*
- *NAS and External Protection*
- *Endpoint Features*
- *Cloud and Virtualization*
- *Databases*

Go back to *About Bacula Enterprise*.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## 1.3 Bacula Users and Administrators

When considering enterprise backup and recovery environments, it is often useful to distinguish between different types or classes of people interacting with the backup and recovery tool:

- Administrators - can control all aspects of Bacula, and modify its configuration.
- Operators - interact with Bacula, following defined procedures, are responsible for certain operational aspects, but do not touch the configuration.
- Users or end-users - people who have no access to the Bacula configuration or all Bacula's features, but may access certain of its functions. A typical example is that a user can restore their own data, to their own computer, but cannot see other backup data or access computers for which they are not responsible.

Bacula itself does not have the concept of users or administrators, but has Consoles that are designed to allow some users to have limited permissions. However, the BWeb Management Suite - the Bacula Enterprise Web-based tool for management and configuration - includes the concepts of users, groups and permissions.

### See also:

Go to:

- *Bacula Enterprise Architecture*
- *Bacula Enterprise Explained*

Go back to the *Bacula Enterprise Fundamentals* chapter.

## 2 Bacula Enterprise Architecture

The following document aims at describing the components of Bacula Enterprise independently, and showing how they work together. It is meant for anybody who wants to have basic understanding of how Bacula is designed.

Bacula Enterprise is made up of the following three core components that work smoothly together:

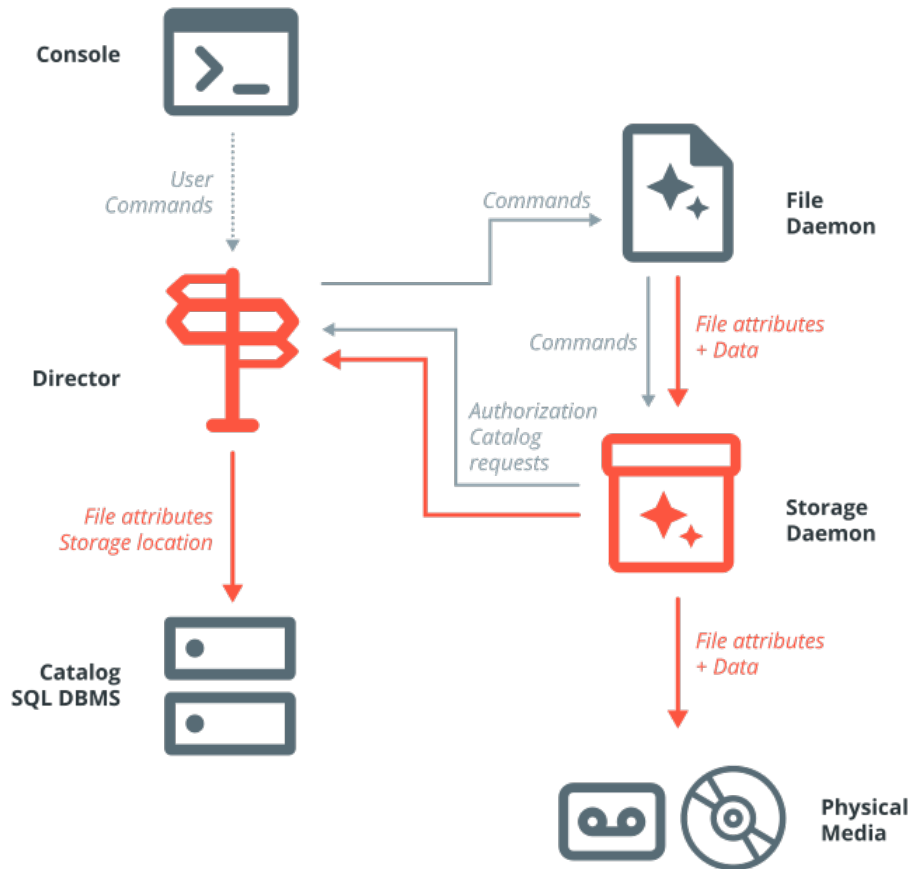


- *Director*
- *Client (File Daemon)*
- *Storage Daemon.*

What is more, vital part of the Bacula architecture are also components/services such as:

- *Catalog*
- *Console*
- *Plugins.*

The flow between particular components is quite simple.



When there is a need to perform any operation in Bacula, one can access the Console and order it. The Console informs the Director about the requested operation, e.g. backup. The Director, being the main Bacula server, schedules and directs the operation. It contacts the chosen Client (File Daemon), and points to the files to be backed up and informs it where it will be saved. It also communicates with the Storage Daemon and suggests where the SD should back up the chosen data and file attributes to *Pools* and *Volumes*. The Client sends the data to the Storage Daemon. At the same time, completely automatically, the Catalog is updated with the information about the operation.

The Bacula Architecture also contains *Plugins*. This component is strictly connected to the File Daemon, or Storage Daemon as the plugins are added to the File/Storage Daemons. They facilitate the backup operation of certain files, e.g. the ones that are managed by another application.

## 2.1 Fundamentals: Catalog

The Catalog (Bacula database) service is responsible for maintaining the bacula database for all files backed up. The Catalog services allow system administrators or users to quickly locate and restore any desired file. The Catalog services sets Bacula apart from simple archiver programs like tar and dump, because the Catalog maintains a record of all *Volumes* used, all *Jobs* run, and all Files saved, ensuring efficient restoration and Volume management.

Bacula can run with three SQL database backends: MySQL, PostgreSQL and SQLite. We strongly recommend using SQLite only for testing purposes. The two others, MySQL and PostgreSQL, provide quite a number of features, including rapid indexing, arbitrary queries, and security.

---

**Note:** PostgreSQL is strongly recommended, and the only Catalog database engine supported with new Bacula Enterprise installations.

---

To keep the Catalog to a manageable size, the backup information should be removed from the Catalog after the defined File *Retention Period*. Bacula provides the mechanisms for the Catalog to be automatically pruned according to the retention periods defined.

**See also:**

Go back to:

- *Director*
- *Client (File Daemon)*
- *Storage Daemon*

Go to:

- *Console*
- *Plugins*

Go back to the *Bacula Enterprise Architecture* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## 2.2 Fundamentals: Client (File Daemon)

The Bacula Client service (also known as the File Daemon) is a software program that is installed on each machine to be backed up. It is specific to the operating system on which it runs, and is responsible for providing file attributes and data when requested by the Director. The Client services are also responsible for the filesystem-dependent part of restoring the file attributes and data during a recovery operation. This program runs as a daemon on the machine to be backed up.



Bacula Enterprise Clients, distributed as binary packages, are available for a wide variety of Operating Systems:

- Windows
- GNU/Linux based platforms
- MacOS
- FreeBSD
- Solaris
- HP-UX
- AIX
- and many more.

Open a [support ticket](#) if you are looking to backup a platform you don't find documented.

Consult the [TechnicalReferenceFileDaemon](#) chapter to know more about the Client resource and all possible directives to configure it.

**See also:**

Go back to:

- *Director*

Go to:

- *Storage Daemon*
- *Fundamentals: Catalog*
- *Console*
- *Plugins*

Go back to the *Bacula Enterprise Architecture* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## 2.3 Fundamentals: Console

The Bacula Console is a program that allows system administrators or users to communicate with the Bacula Director. Bacula Consoles are available with different user front ends: text-based console interface, graphical user interface, and web interface. The first and simplest is to run the **bconsole** program in a shell window (i.e., TTY interface). Also, the most complete tool being **BWeb Management Suite** available in the Bacula Enterprise version.

If you want to give access to a particular user (not overall Bacula administrator), you need to configure a Console for them. In particular, it may be desirable to implement specific Access Control Lists to prevent users from accessing data they are not authorized for. This part is covered in the Console Configuration chapter.

---

**Note:** If you wish to learn about BWeb authentication methods, click [here](#).

---

**See also:**

Go back to:

- *Director*
- *Storage Daemon*
- *Fundamentals: Catalog*

Go to:

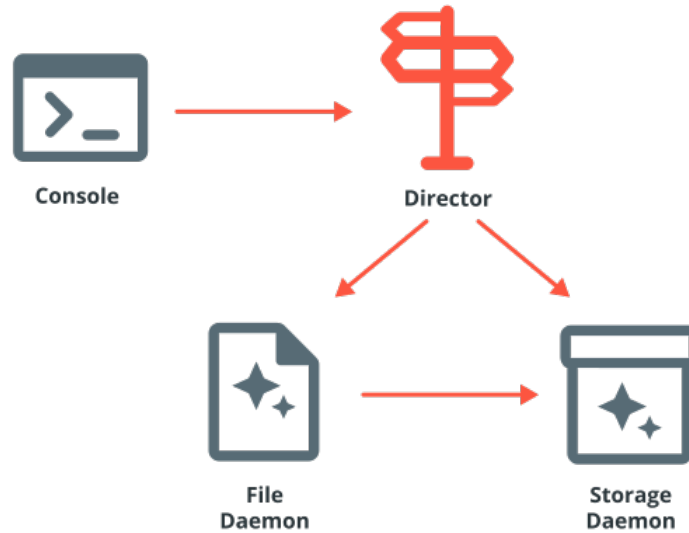
- *Plugins*

Go back to the *Bacula Enterprise Architecture* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

## 2.4 Fundamentals: Director

The Bacula Director is a program that supervises all the backup, restore and verify operations. System administrator use the Bacula Director to schedule backups and to recover files. The Director runs as a daemon service in the background. Bacula Enterprise users access the Director to do backups or restores of their files through a Console like **BWeb** or **bconsole**.



### See also:

Go to:

- *Client (File Daemon)*
- *Storage Daemon*
- *Fundamentals: Catalog*
- *Console*
- *Plugins*

Go back to the *Bacula Enterprise Architecture* chapter.

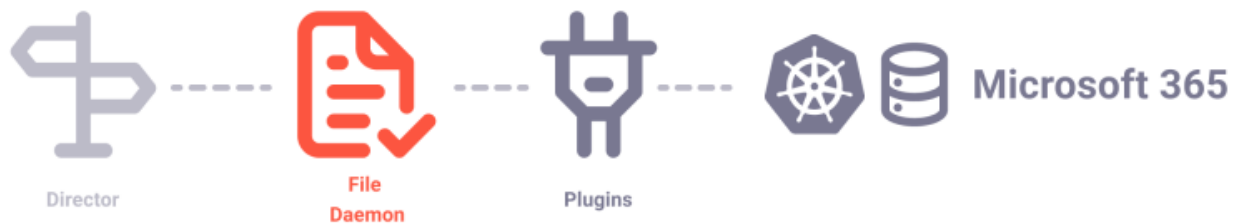
Go back to the *Bacula Enterprise Fundamentals* chapter.

## 2.5 Fundamentals: Plugins

There are two kinds of plugins: one related to the Bacula Storage Daemon, and the other for the Bacula File Daemon. You can distinguish between them by their names:

- a storage daemon plugin is named `xxxx-sd.so`
- a file daemon plugin is named `yyyy-fd.so`

Be sure to install the plugin in the right place (on the File Daemon or the Storage Daemon), and read the related documentation and to set it up correctly before use.



Every plugin is dedicated to a specific set of technologies, usually to an enterprise-grade solution such as PostgreSQL, Kubernetes, or VMware. By default, when working without special plugins, Bacula backs up files and nothing else. To learn more about plugins, look here.

---

**Note:** To see the list of available plugins, visit [Bacula Enterprise Dedicated Backup Solutions](#).

---

**See also:**

Go back to:

- [Director](#)
- [Client \(File Daemon\)](#)
- [Storage Daemon](#)
- [Fundamentals: Catalog](#)
- [Console](#)

Go back to the [Bacula Enterprise Architecture](#) chapter.

Go back to the [Bacula Enterprise Fundamentals](#) chapter.

## 2.6 Fundamentals: Storage Daemon

The Bacula Storage service is a software program that performs the storage and recovery of file attributes and data to the physical backup media or volumes. In other words, the Storage Daemon is responsible for reading and writing your tapes (or other storage media, e.g., files). The Storage service runs as a daemon on the machine that has the backup device (for example a tape drive or a large disk).



The Storage Daemon can write to multiple storage types:

- its local storage
- any attached disk through Fiber Channel, iSCSI, SAN, etc.
- tapes
- clouds.



The data written can be compressed, encrypted or deduplicated. Consult the `StorageDaemonResourceTypes` chapter to know more about the Storage Daemon resource and all possible directives to configure it.

**See also:**

Go back to:

- *Director*
- *Client (File Daemon)*

Go to:

- *Fundamentals: Catalog*
- *Console*
- *Plugins*

Go back to the *Bacula Enterprise Architecture* chapter.

Go back to the *Bacula Enterprise Fundamentals* chapter.

**See also:**

Go back to:

- *About Bacula Enterprise*

Go to:

- *Bacula Enterprise Explained*

Go back to the *Bacula Enterprise Fundamentals* chapter.

## 3 Bacula Enterprise Explained

The following articles aims at presenting the nomenclature regarding Bacula solution, familiarizing the reader with the main concepts of Bacula Enterprise, and explaining the logic behind the solution.

### 3.1 Main Concepts

#### Fundamentals: Jobs

The Job is a basic unit in Bacula, and is run by the Director. Each Job resource definition contains the name of a Client and a FileSet, a Schedule for the Job, location for data to be stored, and specification for *Pools* of *Volumes* to be used. In effect, each Job resource must specify:



## Who

In Bacula's terminology is the Client or the machine that is to be backed up.

See the `TechnicalReferenceDirClientResource` chapter to know more about the Client.

## What

FileSet or the list of files to include and/or exclude.

A FileSet resource is required for each backup Job. It consists of a list of files or directories to be included, a list of files or directories to be excluded and various backup options such as compression, encryption, and signatures that are to be applied to each file.

To get more details, see the `TechnicalReferenceDirFileSetResource` chapter.

## Where

It is defined by the Storage, Pool, and Catalog, where the Storage specifies on what physical device to backup the files, the Pool defines on what specific Volume, and the Catalog specifies where to keep track of it all.

Bacula Enterprise supports the following type of storage:

- Local disk on the host running the SD
- Local file system via remote storage (SAN, NFS)
- Cloud Storage (Azure, Oracle, GCP, Private clouds, Amazon among others)
- Deduplicated storage (ZFS, NetApp, DELL/EMC, HPE among others) via the Aligned plugin
- Local storage or NFS storage with the Global Endpoint Deduplication plugin
- Tape libraries supported by the Linux Kernel through mt
- FiFo

## Director Storage

Each Storage resource is configured in the Director, and it defines the Storage that can be used in Jobs. Each Director Storage resource points to an Autochanger or individual device on a Storage Daemon server.

To get more details, see the `TechnicalReferenceDirStorageResource` chapter.

## Storage Daemon Autochangers

In Bacula, SD Autochanger resources may contain one or more tape or drive devices. Autochangers are a way to group multiple devices into one resource. Jobs can be configured to use an Autochanger, and Bacula will automatically choose the tape or drive device that will be used to read or write the data. When multiple concurrent jobs are using the same Autochanger, Bacula is able to spread the jobs among the available devices in the Autochanger.

To get more details, see the `TechnicalReferenceSDAutochangerResource` chapter.

## Storage Daemon Device

The SD Device resource allows configuring a device used for writing to and reading from backup volumes. The Device can point to a tape drive or directory on the filesystem. Multiple Devices can be grouped into an Autochanger.

To get more details, see the `TechnicalReferenceSDDeviceResource` chapter.

## When

It is defined by what Schedule is used.

The Schedule resource provides a means of automatically starting Jobs at specific intervals.

To get more details, see the [TechnicalReferenceDirScheduleResource](#) chapter.

## Types of Jobs

Those jobs are defined in the Job Resource. There are several types of jobs in Bacula:

- Backup
- Restore
- Admin
- Verify
- Copy
- Migration

### See also:

Go to:

- [Pools](#)
- [Pruning](#)
- [Purging](#)
- [Retention Periods](#)
- [Volumes](#)
- [Truncation](#)

Go back to [Bacula Enterprise Explained](#)

Go back to the [Bacula Enterprise Fundamentals](#) chapter.

## Fundamentals: Pools

The Pool resource defines the set of storage *Volumes* (tapes or files) to be used by Bacula to write the data. It permits to configure the data stored on disk volumes, tapes or cloud storage. By configuring different Pools, you can determine which set of *Volumes* (media) receives the backup data. This permits, for example, to store all full backup data on one set of Volumes into a specific Pool, and all incremental backups on another set of Volumes belonging to another Pool. Alternatively, you could assign a different set of Volumes to each machine that you backup. Another important aspect of a Pool is that it contains the default attributes (Maximum Jobs, Retention Period, Recycle flag, etc.) that will be given to a Volume when it is created. The Pool resource is a key part of a backup strategy as it permits to group data under the same backup policy under the same Pool.

See the [TechnicalReferenceDirPoolResource](#) chapter to know more about the Pools.

See the [Volumes](#) chapter to know more about Volumes.

As mentioned, Pools in Bacula are a way to manage volumes and to manage collections of similar volumes.

---

**Note:** You might have either different requirements or even no requirement to separate volumes into several pools.

---

The examples of Pools:

- Pools: “short-term” and “long-term” to handle short and long retention periods.
- Pools: “big-customer” and “usual-customers” to keep all the backup jobs from your client “Big Customer” in a specific pool while putting all “usual” customers into another one.
- Pools: “r-and-d” and “accounting” to keep R&D backups together and the Accounting ones in another pool, separated from each other.

There can be many reasons to use or not use several Pools. Bacula configuration gives the Backup Administrator the ability to match the defined requirements.

Due to how Bacula manages Volume *retention periods*, the jobs put into the same pool should have the same retention periods.

See more details on how to convert an enterprise backup policy into a Bacula backup policy [here](#).

## Pool Types

Bacula has different kinds of pools:

- Scratch
- Backup
- Recycle

A **Scratch** pool contains volumes that may be with any pool. If no volumes are available, Bacula will look in the Scratch pool and move/supply a volume to that pool.

A **Backup** pool contains volumes intended to keep data from backups for a defined retention period. It can also include Recycled or Purged volumes, i. e. Volumes that are eligible to be overwritten.

A **Recycle** pool contains volumes that have been used in a Backup Pool after the purge process freed them

The Bacula configuration allows the Administrator to attach a Scratch Pool and a Recycle Pool to a Backup pool.

**See also:**

Go back to:

- [Jobs](#)

Go to:

- [Pruning](#)
- [Purging](#)
- [Retention Periods](#)
- [Volumes](#)
- [Truncation](#)

Go back to [Bacula Enterprise Explained](#)

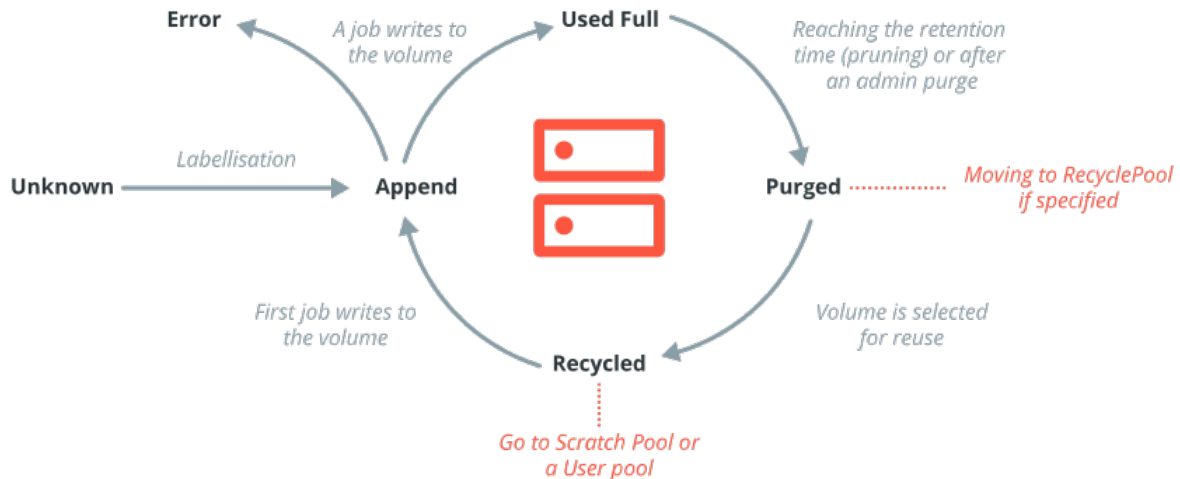
Go back to the [Bacula Enterprise Fundamentals](#) chapter.

## Fundamentals: Volumes

Volume is an archive unit where Bacula stores backed up data, normally a tape or a named disk file where Bacula stores the data from one or more backup jobs. When backing up to tapes, a Volume is identical to a tape and when backing up to disks, a Volume is a file.

Bacula manages all its Volumes in *Pools*. In particular, a Job is not configured to write to any particular Volume, but to a set of Volumes called a Pool. Volumes of Bacula are always members of only one Pool. There can however be multiple pools.

The volume cycle management process is shown in the figure below:



The Volume retention period is defined as the time Bacula will not overwrite a Volume after it was last written to. Therefore, if you write to a volume with a retention period of 20 days once, the volume will not be overwritten for at least 20 days. If you write to it seven days in a row, the volume will not be overwritten for at least 27 days. Remember that Bacula will try to keep your data safe for as long as possible.

## Storage Capacity Management

If you want to overwrite a backup, you need to overwrite the corresponding volume(s). Additionally, to be able to reuse a Volume, its retention period should expire first.

As we already noted, Bacula will try to keep your data safe as long as possible. This means that you can be in a situation (disk backups only) where your storage space is too small even if you defined correct retention periods. In such cases, you can use the **ActionOnPurge** directive requesting Bacula to truncate disk volumes and thus free storage space. For this to work, the bconsole 'truncate' command needs to be scheduled in an Admin job.

## Limiting the Volume Size

### Tapes

On tapes, we usually don't need to explicitly limit the volume size, because the volume is identical to the tape which has a limited and defined maximum capacity.

## Disks

On disks, on the other hand, as volumes are handled with files, there is a risk of filling the disk with only one volume, so we need to limit the volume size and number of volumes. We explained above how the volume retention periods are handled by Bacula. However, often we need to reuse our volumes as soon as possible to be able to reuse the storage space they are using during their retention periods. In the previous case described under the image, we should have restricted the use duration of the Volume.

There are several directives available to limit the occupied storage space in a Pool: `Maximum Volumes`, `Maximum Volumes Bytes`, etc. To browse through the directives, visit: [TechnicalReferenceDirPoolResource](#).

These limitations can be managed manually (by the Administrator with console commands) or automatically (for instance with scripts or Admin jobs).

## Volumes and Pool Modifications

Because of its great flexibility, Bacula can be puzzling. Let's say you defined `Maximum Volume Bytes` to 10 GB and then realize that you want to move it to 15 GB. Then, you will modify your "Pool" resource definition. This is correct and will work for all new volumes created into the Pool. But the existing volumes are not modified by this change. To update the existing volumes already in the Pool with the new values you must update the existing volumes' meta-data.

In `bconsole`, enter 'update volume' and answer the subsequent questions, or alternatively 'update volume allfrompool=`the-pool-name`', or use one of the graphical interfaces to update the existing volumes.

### See also:

Go back to:

- [Jobs](#)
- [Pools](#)
- [Pruning](#)
- [Purging](#)
- [Retention Periods](#)

Go to:

- [Truncation](#)

Go back to [Bacula Enterprise Explained](#)

Go back to the [Bacula Enterprise Fundamentals](#) chapter.

## Fundamentals: Retention Periods

There are various kinds of retention periods that Bacula recognizes. The most important are File Retention Period, Job Retention Period, and the Volume Retention Period. Each of these retention periods applies to the time that specific records will be kept in the Catalog database. This should not be confused with the time that the data saved to a Volume is valid and available for restore – in many cases, data will be available much longer than any of the Retention Periods configured (the data remains in the Volume until the Volume is recycled or truncated).

More details about Schedules and Retentions.

## File Retention Period

File Retention Period determines the time that File records are kept in the Catalog database. This period is important. As long as File records remain in the database, you can “browse” the database with a Console program and restore any individual file. Once the files are removed or pruned from the database, these files associated with a backup job can no longer be browsed. File records of a Volume use the most space in the database. As a consequence, you must ensure that regular pruning of the file records is done to keep your database from growing too large.

See the Console prune command for more details on this subject.

## Job Retention Period

Job Retention Period is the length of time that Job records will be kept in the database. Note that Files records are associated to a job. File Records can also be purged disassociating themselves from a job. In this case, information will be available about the Jobs that ran, but not the details of the files that were backed up. Normally, when a job record is purged, all of its associated file records will also be purged.

## Volume Retention Period

Volume Retention Period is the length of time between the last write and the volume’s re-usage moment. Simply, it’s the period of the files being stored. Bacula will normally never overwrite a Volume that contains the only backup copy of a file. As for the Catalog, it retains information for all files backed up for all current Volumes. Once the Volume is overwritten, data from the Catalog is removed as well. Thus, if there is a very large Pool of Volumes or a Volume is never overwritten, the Catalog database may become enormous. Bacula will normally not overwrite a Volume that contains data still inside its Retention period, but select a Volume which is out of its Retention time for recycling.

### See also:

Go back to:

- [Jobs](#)
- [Pools](#)
- [Pruning](#)
- [Purging](#)

Go to:

- [Volumes](#)
- [Truncation](#)

Go back to [Bacula Enterprise Explained](#)

Go back to the [Bacula Enterprise Fundamentals](#) chapter.

## Fundamentals: Pruning

To keep the Catalog to a manageable size, the backup information should be removed (pruned) from the Catalog after the defined File and Job Retention Periods. Bacula by default automatically prunes the Catalog database entries according to the retention periods you define.

The autopruning chapter will give you more details about the pruning process and will assist you to choose between manual pruning and automatic pruning.

Pruning by default occurs at the end of a Job. When doing some tests or if you only have few jobs a day, this could be fine. But as soon as the number of jobs is growing, you would prefer to manage pruning another way to let your Catalog do its best for the backup jobs, not for the database administrative tasks. For more details, [click here](#).

### See also:

Go back to:

- [Jobs](#)
- [Pools](#)

Go to:

- [Purging](#)
- [Retention Periods](#)
- [Volumes](#)
- [Truncation](#)

Go back to [Bacula Enterprise Explained](#)

Go back to the [Bacula Enterprise Fundamentals](#) chapter.

## Fundamentals: Purging

Once all the database records that concern a particular Volume have been pruned, respecting the retention periods, the Volume is said to be purged (i.e. has no more catalog entries).

It is, however, possible to submit commands to Bacula to purge specific information, which will not respect configured retention periods. Naturally, this is something that should only be done with the greatest care.

Bacula will try to keep your data safe as long as possible, thus purging a Volume will not automatically reclaim the used space. If you want to reuse space, you must configure Bacula accordingly.

See the Console purge command for more details on this subject.

### See also:

Go back to:

- [Jobs](#)
- [Pools](#)
- [Pruning](#)

Go to:

- [Retention Periods](#)
- [Volumes](#)
- [Truncation](#)



Go back to *Bacula Enterprise Explained*

Go back to the *Bacula Enterprise Fundamentals* chapter.

## **Fundamentals: Truncation**

Truncation erases the Volume and its data from disk. Therefore, there is no possibility to recover it. Unlike purging which changes the volume status in the Catalog, but does not touch your Volume or its data leaving it on disk.

### **See also:**

Go back to:

- *Jobs*
- *Pools*
- *Pruning*
- *Purging*
- *Retention Periods*
- *Volumes*

Go back to *Bacula Enterprise Explained*

Go back to the *Bacula Enterprise Fundamentals* chapter.

### **See also:**

Go back to :

- *About Bacula Enterprise*
- *Bacula Enterprise Architecture*

Go back to the *Bacula Enterprise Fundamentals* chapter.