

```

#
# Default Bacula Director Configuration file
#
# The only thing that MUST be changed is to add one or more
# file or directory names in the Include directive of the
# FileSet resource.
#
# For Bacula release 5.0.0 (26 January 2010) -- redhat (Final)
#
# You might also want to change the default email address
# from root to your address. See the "mail" and "operator"
# directives in the Messages resource.
#
# Updates and annotation by K. M. Peterson
# http://kmpeterson.com/special/bblisa-bacula
#

Director {
    # define myself
    Name = bacula-dir
    DIRport = 9101 # where we listen for UA connections
    QueryFile = "/usr/libexec/bacula/query.sql"
    WorkingDirectory = "/var/spool/bacula"
    PidDirectory = "/var/run"
    Maximum Concurrent Jobs = 1
    Password = "BaCuLaDiRpAsSw0rD" # Console password
    Messages = Daemon
}

JobDefs {
    # defaults - entries in Job definitions override these
    Name = "DefaultJob"
    Type = Backup
    Level = Incremental # will get forced to "Full" on first run
    Client = bacula-fd # by default, points to file daemon on this host
    FileSet = "Full Set" # pointer to list/specification of files for this backup
    Schedule = "WeeklyCycle" # pointer to a scheduling directive
    Storage = File # references a "device" definition
    Messages = Standard # message logging/destination configuration
    Pool = File # media definition
    Priority = 10
    Write Bootstrap = "/var/spool/bacula/%c.bsr" # recovery information
}

#
# Define the main nightly save backup job
# By default, this job will back up to disk in /tmp
Job {
    Name = "BackupServerFiles"
    JobDefs = "DefaultJob" # simply use defaults
}

Job {
    Name = "BackupClient2"
    Client = bacula-test-y-fd # convention: hostname-FD; client defined below
    JobDefs = "DefaultJob"
}

# Backup the catalog database (after the nightly save)
Job {
    Name = "BackupCatalog"
    JobDefs = "DefaultJob"
    Level = Full
    FileSet="Catalog"
    Schedule = "WeeklyCycleAfterBackup"
    # This creates an ASCII copy of the catalog
    # Arguments to make_catalog_backup.pl are:
    # make_catalog_backup.pl <catalog-name>
    RunBeforeJob = "/usr/libexec/bacula/make_catalog_backup.pl MyCatalog"
    # This deletes the copy of the catalog
    RunAfterJob = "/usr/libexec/bacula/delete_catalog_backup"
    Write Bootstrap = "/var/spool/bacula/%n.bsr"
    Priority = 11 # run after main backup
}

#
# Standard Restore template, to be changed by Console program
# Only one such job is needed for all Jobs/Clients/Storage ...
#
Job {
    Name = "RestoreFiles"
    Type = Restore
    Client=bacula-fd
    FileSet="Full Set"
    Storage = File
    Pool = Default
    Messages = Standard
    Where = /tmp/bacula-restores
}

```

```

# List of files to be backed up
FileSet {
  Name = "Full Set"
  Include {
    Options {
      signature = MD5           # catalog will store digest of file
      compression = GZIP       # catalog will store digest of file
    }
  }
#
# Put your list of files here, preceded by 'File =', one per line
# or include an external list with:
#
#   File = <file-name>
#
# Note: / backs up everything on the root partition.
# if you have other partitions such as /usr or /home
# you will probably want to add them too.
#
# By default this is defined to point to the Bacula binary
# directory to give a reasonable FileSet to backup to
# disk storage during initial testing.
#
  File = /boot                 # critical data
  File = /usr                  # and
  File = /root                 # system
  File = /var                  # files
  File = /home                 # and user directories
}

#
# If you backup the root directory, the following two excluded
# files can be useful
#
Exclude {
  File = /var/spool/bacula     # exclude bacula in-process dat
  File = /var/lib/mysql       # don't back up database files
  File = /var/backup          # don't back up backups et cetera et all ad infinitum
  File = /tmp
  File = /proc
  File = /tmp                 # exclude ... etc.
  File = /.journal
  File = /.fsck
}
}

#
# When to do the backups, full backup on first sunday of the month,
# differential (i.e. incremental since full) every other sunday,
# and incremental backups other days
Schedule {
  Name = "WeeklyCycle"
  Run = Full 1st sun at 23:05
  Run = Differential 2nd-5th sun at 23:05
  Run = Incremental mon-sat at 23:05
}

# This schedule does the catalog. It starts after the WeeklyCycle
Schedule {
  Name = "WeeklyCycleAfterBackup"
  Run = Full sun-sat at 23:10
}

# This is the backup of the catalog
FileSet {
  Name = "Catalog"
  Include {
    Options {
      signature = MD5
    }
  }
  File = "/var/spool/bacula/bacula.sql" # this file created by 'RunBeforeJob'
                                        # directive in BackupCatalog job,
                                        # then deleted by 'RunAfterJob' script.
}

# Client (File Services) to backup
Client {
  Name = bacula-fd             # matches Name directive in FD
  Address = bacula-test-x     # hostname
  FDPort = 9102
  Catalog = MyCatalog
  Password = "bAcU1AfDpAsSwOrDx@" # password for FileDaemon
  File Retention = 30 days      # 30 days
  Job Retention = 6 months     # six months
  AutoPrune = yes             # Prune expired Jobs/Files
}

#
# Second Client (File Services) to backup
# You should change Name, Address, and Password before using

```

```

#
Client {
  Name = bacula-test-y-fd
  Address = bacula-test-y                # hostname of client (hosts file for testing)
  FDPort = 9102
  Catalog = MyCatalog
  Password = "BaCuLaFdPaSsWoRdY"        # password for FileDaemon 2
  File Retention = 30 days               # 30 days
  Job Retention = 6 months                # six months
  AutoPrune = yes                         # Prune expired Jobs/Files
}

# Definition of file storage device
Storage {
  Name = File
  # Do not use "localhost" here
  Address = bacula-test-x                # local host, in this example.
  SDPort = 9103
  Password = "bAcUlAsDpAsSwOrD"
  Device = FileStorage
  Media Type = File
}

# Definition of DDS tape storage device
#Storage {
#  Name = DDS-4
#  Do not use "localhost" here
#  Address = storage.example.com         # N.B. Use a fully qualified name here
#  SDPort = 9103
#  Password = "@@SD_PASSWORD@"          # password for Storage daemon
#  Device = DDS-4                        # must be same as Device in Storage daemon
#  Media Type = DDS-4                    # must be same as MediaType in Storage daemon
#  Autochanger = yes                     # enable for autochanger device
#}

# Definition of 8mm tape storage device
#Storage {
#  Name = "8mmDrive"
#  Do not use "localhost" here
#  Address = storage.example.com         # N.B. Use a fully qualified name here
#  SDPort = 9103
#  Password = "@@SD_PASSWORD@"
#  Device = "Exabyte 8mm"
#  MediaType = "8mm"
#}

# Definition of DVD storage device
#Storage {
#  Name = "DVD"
#  Do not use "localhost" here
#  Address = storage.example.com         # N.B. Use a fully qualified name here
#  SDPort = 9103
#  Password = "@@SD_PASSWORD@"
#  Device = "DVD Writer"
#  MediaType = "DVD"
#}

# Generic catalog service
Catalog {
  Name = MyCatalog
  # Uncomment the following line if you want the dbi driver
  # dbdriver = "dbi:sqlite3"; dbaddress = 127.0.0.1; dbport =
  dbname = "bacula"; dbuser = "bacula"; dbpassword = "BaCuLaCaTaLoG"
}

# Reasonable message delivery -- send most everything to email address
# and to the console
Messages {
  Name = Standard
#
# NOTE! If you send to two email or more email addresses, you will need
# to replace the %r in the from field (-f part) with a single valid
# email address in both the mailcommand and the operatorcommand.
# What this does is, it sets the email address that emails would display
# in the FROM field, which is by default the same email as they're being
# sent to. However, if you send email to more than one address, then
# you'll have to set the FROM address manually, to a single address.
# for example, a 'no-reply@mydomain.com', is better since that tends to
# tell (most) people that its coming from an automated source.
#
# mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\ (Bacula) \<%r\>\" -s \"Bacula: %t %e of %c %l\" %r"
# operatorcommand = "/usr/sbin/bsmtp -h localhost -f \"\ (Bacula) \<%r\>\" -s \"Bacula: Intervention needed
# for %j\" %r"
# mail = root@localhost = all, !skipped

```

```

operator = root@localhost = mount
console = all, !skipped, !saved
#
# WARNING! the following will create a file that you must cycle from
#           time to time as it will grow indefinitely. However, it will
#           also keep all your messages if they scroll off the console.
#
append = "/var/spool/bacula/log" = all, !skipped
catalog = all
}

#
# Message delivery for daemon messages (no job).
Messages {
Name = Daemon
mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\(\Bacula\) \(<%r\>\)" -s \"Bacula daemon message\" %r"
mail = root@localhost = all, !skipped
console = all, !skipped, !saved
append = "/var/log/bacula.log" = all, !skipped
}

# Default pool definition (Unused)
Pool {
Name = Default
Pool Type = Backup
Recycle = yes # Bacula can automatically recycle Volumes
AutoPrune = yes # Prune expired volumes
Volume Retention = 365 days # one year
}

# File Pool definition (This is pool used for jobs defined off JobDefs 'DefaultJob')
Pool {
Name = File
Pool Type = Backup
Recycle = yes # Bacula can automatically recycle Volumes
AutoPrune = yes # Prune expired volumes
Volume Retention = 365 days # one year
Maximum Volume Bytes = 1G # Limit Volume size for ease of maintenance.
Maximum Volumes = 100 # Limit number of Volumes in Pool
Label Format = "Test-" # Filename format (required to auto label!)
}

# Scratch pool definition
Pool {
Name = Scratch
Pool Type = Backup
}

#
# Restricted console used by tray-monitor to get the status of the director
#
Console {
Name = bacula-mon
Password = "BaCuLaMoNdIrPaSsWoRd"
CommandACL = status, .status
}

```