

## How to setup Bacula for a Weekly Backup Job

In this scenario, we have been asked to create a weekly backup job that backups an NFS-mounted folder named /mnt/database. Our Bacula server is a Linux machine.

Since the database directory is modified every day, we will need to do incremental backups as well as a full backup once a week.

So, let's get started and setup this job. I am assuming you've installed Bacula on your system and are using the standard directory for Bacula configuration, /etc/bacula

1. First, we need to create a pool definition for the new job, we will call this one DBPool. In the /etc/bacula/bacula-dir.conf file, add a definition for the pool like this:

```
Pool {  
    Name = DBPool  
    Pool Type = Backup  
    Recycle = yes  
    RecyclePool = Scratch  
    AutoPrune = yes  
    Volume Retention = 3 months  
    Cleaning Prefix = "CLN"  
}
```

Some quick definitions of these options:

Name refers to what you want to call this pool.

Pool Type should always be Backup for backup jobs. There are other types but that is outside the scope of this howto.

Recycle = yes tells Bacula that we can recycle volumes in this pool

RecyclePool, this is where Bacula will look to grab recycled volumes automatically. I set this to Scratch as I have created a Recycle pool named Scratch.

Autoprune allows Bacula to automatically prune jobs and files from this pool after a set amount. This value is defined elsewhere in the bacula-dir.conf file.

Volume Retention, this allows you to set how long to retain volumes before recycling them. In this case, 3 months.

Cleaning Prefix, this is needed so cleaning tapes are not used in this pool.

2. Now that we have our pool defined, we will create a FileSet definition. This tells Bacula where to find the files and sets any specific options for handling the files, (i.e. compression, encryption, etc..) Here's the definition:

```
FileSet {  
    Name = "DB"  
    Include {  
        Options {  
            signature = MD5  
        }  
        File = /mnt/database  
    }  
}
```

There are several other options you can include here, but I'm going to keep it simple and just enable the Signature = MD5 setting which ensures that when restoring files, the file is the correct file. The MD5 hash is stored in the Bacula database and compared when restoring files. If you want to add more options, feel free to reference the Bacula Reference Guide which explains these further.

3. Next we need to set the Schedule definition for this job. This is what tells Bacula when to run this job. Since we've been given the requirement to do this backup daily with a full backup weekly, we can set a definition like this to handle that:

```
Schedule {  
    Name = "DatabaseBackup"  
    Run = Level=Incremental sun-fri at 00:00  
    Run = Level=Full saturday at 00:00  
}
```

This is a basic example, and of course you can tweak the time it runs, but it shows you how to do a basic weekly incremental/full backup.

4. Next we need to establish the Job definitions, or JobDefs for this. These are the default priorities, schedule, device to backup to, etc.. for a job. Here is a simple JobDefs for this sample job:

```
JobDefs {  
    Name = "DBDefs"  
    Type = Backup  
    Client = linuxbackup-fd  
    FileSet = "DB"  
    Schedule = "DatabaseBackup"  
    Storage = TapeLibrary1  
    Messages = Standard  
    Pool = DBPool  
    Priority = 10  
    write Bootstrap = "/var/bacula/working/%c.bsr"  
}
```

Again, these can be customized further, but this is a basic default option set for this job. I like to do put these rarely-changing parameters in a JobDefs definition instead of a Job definition which we'll tackle next.

5. Finally, we need to create the actual Job definition. This is the meat and potatoes for everything. Here's the Job definition I will use for this job:

```
Job {  
    Name = "DB_Backup"  
    Type = Backup  
    JobDefs = "DBDefs"  
    Enabled = yes  
    Maximum Concurrent Jobs = 1  
}
```



[www.spkaa.com](http://www.spkaa.com)  
Ph: 888-310-4540

---

*SPK and Associates*  
900 E Hamilton Ave, Ste. 100  
Campbell, CA 95008

With this Job definition in place, we can load up the Bacula console with the bconsole command, type in reload to reload the bacula-dir.conf file and now we should have a new job set to run at midnight. On it's initial run it will do a full backup to set a baseline, then it will start on the incrementals and weekly full backups.