To include the desired backup schedule with appropriate pools in your Bacula configuration, we'll need to set up the necessary job definitions, schedules, and pools. Here are the updated steps:

1. Install Bacula on the Backup Server

1.1. Update and Install Bacula

bash

```
sudo apt update
sudo apt install bacula-server bacula-client
```

1.2. Configure Bacula Director (`/etc/bacula/bacula-dir.conf`)

Edit the Bacula Director configuration file to define the jobs, file sets, schedules, and pools:

bash

```
sudo nano /etc/bacula/bacula-dir.conf
```

Add the following configurations:

bash

```
# Define the file sets for each server and workstation
FileSet {
  Name = "DataServerSet"
  Include {
    Options {
      signature = MD5
    }
    File = /data
  }
}

FileSet {
  Name = "NFSServerSet"
  Include {
    Options {
      signature = MD5
    }
    File = /media
  }
}

FileSet {
  Name = "ClientSet"
  Include {
    Options {
      signature = MD5
    }
    File = /home
    File = /etc
  }
}

# Define the pools
Pool {
  Name = "DailyPool"
  PoolType = Backup
  Recycle = yes
  AutoPrune = yes
```

```
  Volume Retention = 7 days
  Maximum Volume Bytes = 50G
  Maximum Volumes = 5
}

Pool {
  Name = "WeeklyPool"
  PoolType = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 30 days
  Maximum Volume Bytes = 100G
  Maximum Volumes = 4
}

Pool {
  Name = "MonthlyPool"
  PoolType = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 180 days
  Maximum Volume Bytes = 200G
  Maximum Volumes = 6
}

# Define the schedules
Schedule {
  Name = "DailySchedule"
  Run = Incremental daily at 23:00
}

Schedule {
  Name = "WeeklySchedule"
  Run = Differential 1st sun at 23:05
}

Schedule {
  Name = "MonthlySchedule"
  Run = Full 1st sun at 23:10
}

# Define the jobs for each server and workstation
Job {
  Name = "BackupDataServer"
  JobDefs = "DefaultJob"
  FileSet = "DataServerSet"
  Client = DataServer-fd
  Schedule = "DailySchedule"
  Storage = File
  Messages = Standard
  Pool = DailyPool
}

Job {
  Name = "BackupDataServerWeekly"
  JobDefs = "DefaultJob"
  FileSet = "DataServerSet"
  Client = DataServer-fd
  Schedule = "WeeklySchedule"
  Storage = File
  Messages = Standard
  Pool = WeeklyPool
}
```

```
Job {
  Name = "BackupDataServerMonthly"
  JobDefs = "DefaultJob"
  FileSet = "DataServerSet"
  Client = DataServer-fd
  Schedule = "MonthlySchedule"
  Storage = File
  Messages = Standard
  Pool = MonthlyPool
}

Job {
  Name = "BackupNFSServer"
  JobDefs = "DefaultJob"
  FileSet = "NFSServerSet"
  Client = NFSServer-fd
  Schedule = "DailySchedule"
  Storage = File
  Messages = Standard
  Pool = DailyPool
}

Job {
  Name = "BackupNFSServerWeekly"
  JobDefs = "DefaultJob"
  FileSet = "NFSServerSet"
  Client = NFSServer-fd
  Schedule = "WeeklySchedule"
  Storage = File
  Messages = Standard
  Pool = WeeklyPool
}

Job {
  Name = "BackupNFSServerMonthly"
  JobDefs = "DefaultJob"
  FileSet = "NFSServerSet"
  Client = NFSServer-fd
  Schedule = "MonthlySchedule"
  Storage = File
  Messages = Standard
  Pool = MonthlyPool
}

Job {
  Name = "BackupClient"
  JobDefs = "DefaultJob"
  FileSet = "ClientSet"
  Client = Client-fd
  Schedule = "DailySchedule"
  Storage = File
  Messages = Standard
  Pool = DailyPool
}

Job {
  Name = "BackupClientWeekly"
  JobDefs = "DefaultJob"
  FileSet = "ClientSet"
  Client = Client-fd
  Schedule = "WeeklySchedule"
  Storage = File
  Messages = Standard
  Pool = WeeklyPool
```

```
}

Job {
  Name = "BackupClientMonthly"
  JobDefs = "DefaultJob"
  FileSet = "ClientSet"
  Client = Client-fd
  Schedule = "MonthlySchedule"
  Storage = File
  Messages = Standard
  Pool = MonthlyPool
}

# Define the clients
Client {
  Name = DataServer-fd
  Address = dataserver.example.com
  FDPort = 9102
  Catalog = MyCatalog
  Password = "DataServerPassword"
  File Retention = 60 days
  Job Retention = 6 months
  AutoPrune = yes
}

Client {
  Name = NFSServer-fd
  Address = nfsserver.example.com
  FDPort = 9102
  Catalog = MyCatalog
  Password = "NFSServerPassword"
  File Retention = 60 days
  Job Retention = 6 months
  AutoPrune = yes
}

Client {
  Name = Client-fd
  Address = client.example.com
  FDPort = 9102
  Catalog = MyCatalog
  Password = "ClientPassword"
  File Retention = 60 days
  Job Retention = 6 months
  AutoPrune = yes
}
```

1.3. Configure Bacula Storage Daemon (`/etc/bacula/bacula-sd.conf`)

Edit the Bacula Storage Daemon configuration file to define the storage device:

bash

```
sudo nano /etc/bacula/bacula-sd.conf
```

Add the following configuration:

bash

```
Device {
  Name = FileStorage
  MediaType = File
  ArchiveDevice = /mnt/backup
```

```
  LabelMedia = yes
  RandomAccess = yes
  AutomaticMount = yes
  RemovableMedia = no
  AlwaysOpen = no
}
```

1.4. Configure Bacula File Daemon (`/etc/bacula/bacula-fd.conf`)

The File Daemon configuration on the backup server should remain as default.

1.5. Start Bacula Services

bash

```
sudo systemctl start bacula-dir
sudo systemctl start bacula-sd
sudo systemctl start bacula-fd

sudo systemctl enable bacula-dir
sudo systemctl enable bacula-sd
sudo systemctl enable bacula-fd
```

2. Install and Configure Bacula File Daemon on Data Server

2.1. Install Bacula Client

bash

```
sudo apt update
sudo apt install bacula-client
```

2.2. Configure Bacula File Daemon (`/etc/bacula/bacula-fd.conf`)

Edit the Bacula File Daemon configuration file:

bash

```
sudo nano /etc/bacula/bacula-fd.conf
```

Add the following configuration:

bash

```
FileDaemon {
  Name = DataServer-fd
  FDport = 9102
  WorkingDirectory = /var/lib/bacula
  PidDirectory = /var/run/bacula
  MaximumConcurrentJobs = 20
}

# Director configuration
Director {
  Name = bacula-dir
  Password = "DataServerPassword"
}

# Messages configuration
Messages {
  Name = Standard
```

```
  director = bacula-dir = all, !skipped, !restored
}
```

2 vulnerabilities detected

Hardcoded Credentials

Embedding credentials in source code risks unauthorized access

Hardcoded Credentials

Embedding credentials in source code risks unauthorized access

2.3. Start Bacula File Daemon Service

bash

```
sudo systemctl start bacula-fd
sudo systemctl enable bacula-fd
```

3. Install and Configure Bacula File Daemon on NFS Server

3.1. Install Bacula Client

bash

```
sudo apt update
sudo apt install bacula-client
```

3.2. Configure Bacula File Daemon (`/etc/bacula/bacula-fd.conf`)

Edit the Bacula File Daemon configuration file:

bash

```
sudo nano /etc/bacula/bacula-fd.conf
```

Add the following configuration:

bash

```
FileDaemon {
  Name = NFSServer-fd
  FDport = 9102
  WorkingDirectory = /var/lib/bacula
  PidDirectory = /var/run/bacula
  MaximumConcurrentJobs = 20
}

# Director configuration
Director {
  Name = bacula-dir
  Password = "NFSServerPassword"
}

# Messages configuration
Messages {
  Name = Standard
  director = bacula-dir = all, !skipped, !restored
}
```

3.3. Start Bacula File Daemon Service

bash

```
sudo systemctl start bacula-fd
sudo systemctl enable bacula-fd
```

4. Install and Configure Bacula File Daemon on Client Workstation

4.1. Install Bacula Client

bash

```
sudo apt update
sudo apt install bacula-client
```

4.2. Configure Bacula File Daemon (`/etc/bacula/bacula-fd.conf`)

Edit the Bacula File Daemon configuration file:

bash

```
sudo nano /etc/bacula/bacula-fd.conf
```

Add the following configuration:

bash

```
FileDaemon {
  Name = Client-fd
  FDport = 9102
  WorkingDirectory = /var/lib/bacula
  PidDirectory = /var/run/bacula
  MaximumConcurrentJobs = 20
}

# Director configuration
Director {
  Name = bacula-dir
  Password = "ClientPassword"
}

# Messages configuration
Messages {
  Name = Standard
  director = bacula-dir = all, !skipped, !restored
}
```

2 vulnerabilities detected

Hardcoded Credentials

Embedding credentials in source code risks unauthorized access

Hardcoded Credentials

Embedding credentials in source code risks unauthorized access

4.3. Start Bacula File Daemon Service

bash

```
sudo systemctl start bacula-fd
sudo systemctl enable bacula-fd
```

5. Verify the Configuration

5.1. Test Connectivity

Ensure that the Bacula Director can communicate with all File Daemons. Use the `bconsole` command on the backup server to check the status of each client:

bash

```
sudo bconsole
```

Once in the Bacula console:

bash

```
status client=DataServer-fd
status client=NFSServer-fd
status client=Client-fd
```

5.2. Run a Manual Backup

You can manually start a backup job to verify that everything is working correctly:

bash

```
run job=BackupDataServer
run job=BackupNFSServer
run job=BackupClient
```

Follow the prompts to start the backup jobs.

Summary

- **Backup Server**: Install and configure Bacula Director, Storage Daemon, and File Daemon.
- **Data Server, NFS Server, Client Workstation**: Install and configure Bacula File Daemon.
- **Configuration**: Define jobs, file sets, and clients in Bacula Director configuration.
- **Verification**: Test connectivity and run manual backups to ensure everything is set up correctly.

This guide provides a comprehensive setup for Bacula on a client-server network, ensuring that critical directories on each server and workstation are backed up efficiently.