Backing up Nextcloud

Consistent Backup and Restore of Application Data

Sebastian Lederer

dass IT GmbH

WHAT IS THIS ABOUT?

- Backing up nextcloud as an example for backing up applications
- nextcloud is a php application using files and MySQL database

HOW TO DO IT?

- Bareos has some features to deal with applications
- multiple solutions are possible
- increasingly complexity levels

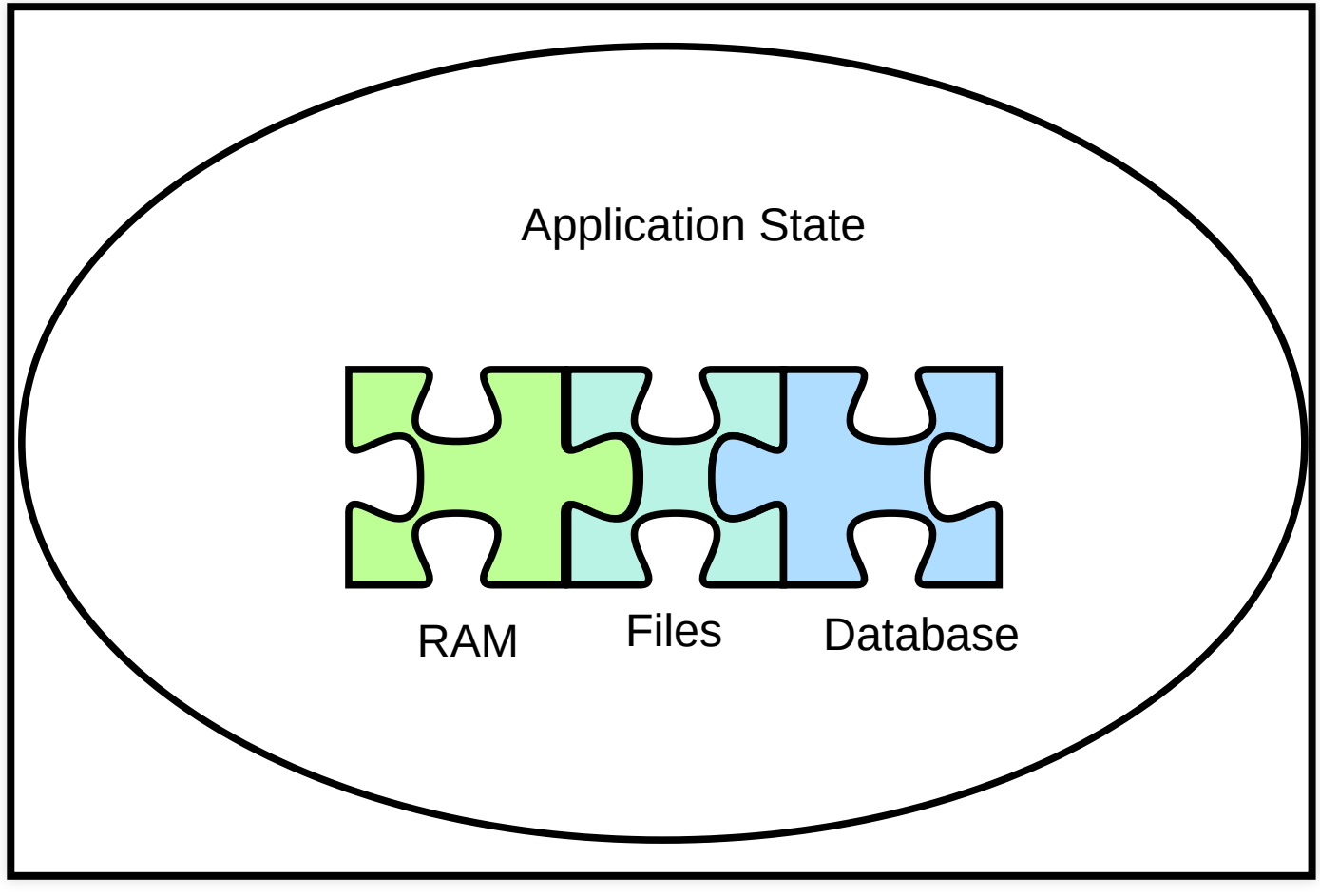
HOW TO DO IT?

- Bareos has some features to deal with applications
- multiple solutions are possible
- increasingly complexity levels
- most complex solution might not be the best

HOW TO DO IT?

- Bareos has some features to deal with applications
- multiple solutions are possible
- increasingly complexity levels
- most complex solution might not be the best
- want reliability and ease of use over fancy features

WHY DO WE NEED THIS?



internal state needs to be backed up consistently

WHAT ABOUT NEXTCLOUD?

- stores uploaded files directly in the filesystem (`/var/www/nextcloud`)
- has a database which stores metadata, e.g. what files are there
- if we restore a file, it is not visible in nextcloud

WORKAROUND IN NEXTCLOUD

There is a nextcloud command line utility which searches for files and adds them to the database:

```
occ files:scan <username>
```

Let's start!

LEVEL 0: DO NOTHING

- just back up all files and hope for the best
- usually a bad idea
- might work for some applications, e.g. Postfix SMTP server

LEVEL 1: SHUT DOWN APPLICATION

- simplest solution
- shut down the application before backup (and restore)
- interrupts service, for minutes or hours
- can be faster than more complicated solutions
- might be the only option for some applications

LEVEL 1: SHUT DOWN APPLICATION

- Create a shell script or some command to run before/after backup
- Could be done just via cron
- Bareos can be used for this, but next solution

LEVEL 2: EXPORT DATA

- export application data to a file before backup runs, then back up this file
- typical example: `mysqldump`
- does not necessarily interrupt service, may slow it down
- job takes longer: export has to run first
- export data takes up extra space

LEVEL 2: CONFIGURATION

Bareos uses the following directives to run extra commands:

`RunBeforeJob`, `RunAfterJob`, `ClientRunBeforeJob`, `ClientRunAfterJob`

LEVEL 2: CONFIGURATION

These are part of the Job resource:

```
Job {
  Name = "backup-nextcloud"
  JobDefs = "DefaultJob"
  Client = "ubuntu-fd"
  FileSet = "nextcloud"
  ClientRunBeforeJob = "/usr/local/sbin/nextcloud_before.sh"
  ClientRunAfterJob = "/usr/local/sbin/nextcloud_after.sh"
}
```

LEVEL 2: CONFIGURATION

You will need a special restore Job:

```
Job {
  Name = "restore-nextcloud"
  Description = "restore nextcloud files and database"
  Type = Restore
  Client = ubuntu-fd
  FileSet = "nextcloud"
  Storage = File
  Pool = Incremental
  Messages = Standard
  Where = /
  ClientRunAfterJob = "/usr/local/sbin/nextcloud_after_restore.sh
}
```

LEVEL 2: CONFIGURATION

The scripts `nextcloud_before.sh` look like this:

```
#!/bin/sh
dumpdir=/var/tmp/nextcloud-backup
dumpfile=$dumpdir/mysqldump.sql

if [ ! -d  "$dumpdir" ]
then
        mkdir "$dumpdir"
fi

/usr/local/sbin/occ maintenance:mode --on

mysqldump --all-databases --single-transaction >$dumpfile
```

LEVEL 2: CONFIGURATION

The `nextcloud_after_restore.sh` script is even simpler:

```
#!/bin/sh
mysql </var/tmp/nextcloud-backup/mysqldump.sql
/usr/local/sbin/occ maintenance:mode --off
```

LEVEL 3: STREAMING DATA

- same as before except writing export data into a pipe
- uses the `bpipe` Bareos plugin
- script has to write/read everything to/from `stdout/stdin`
- saves time, export runs in parallel to backup
- needs no extra storage

LEVEL 3: CONFIGURATION

- plugin needs to be configured in the file daemon
- call the plugin from the `FileSet` resource

```
FileSet {
  Name = "nextcloud-bpipe"
  Description = "Backup nextcloud dirs and mysql dump"
  Include {
    Options {
      Signature = MD5 # calculate md5 checksum per file
      Compression = GZIP
    }
    File = /var/www/nextcloud
    Plugin = "bpipe:file=/MYSQL/dump.sql:reader=/usr/local/sbin/n
  }
}
```

LEVEL 3: CONFIGURATION

Configuration for the file daemon: `/etc/bareos/bareos-fd.d/client/myself.conf`

```
Client {
  Name = ubuntu-fd
  Plugin Directory = "/usr/lib/bareos/plugins"
  Plugin Names = "bpipe"
}
```

LEVEL 4: APPLICATION SPECIFIC PLUGIN

- some applications have their own plugin
- you can write your own plugin in python using the python plugin
- for nextcloud, bareos_percona plugin (works with MySQL/MariaDB)
- data is stored as virtual files on backup volumes
- allows incremental database backups
- needs complex script to restore database

LEVEL 4: CONFIGURATION

we are using `xtrabackup` from Percona:

https://www.percona.com/doc/percona-xtrabackup/LATEST/installation.html#installing-percona-xtrabackup-from-repositories

LEVEL 4: CONFIGURATION

bareos_percona Plugin is not part of main bareos packages, comes from bareos-contrib repository

Install it like this:

```
git clone https://github.com/bareos/bareos-contrib.git
cp bareos-contrib/fd-plugins/bareos_percona/*.py /usr/lib/bareos/
```

LEVEL 4: CONFIGURATION

`bareos_percona` plugin currently does not work with Bareos 18.04, trivial patch

```
--- BareosFdWrapper.py.orig     2018-09-19 16:55:37.793758890 +0000
+++ BareosFdWrapper.py  2018-09-19 17:17:38.759084496 +0000
@@ -62,7 +62,7 @@
     return bareos_fd_plugin_object.plugin_io(context, IOP)


-def create_file(context, restorepkt):
+def CreateFile(context, restorepkt):
     return bareos_fd_plugin_object.create_file(context, restorepkt)
```

LEVEL 4: CONFIGURATION

Things to consider with the `bareos_percona` plugin:

A restore does not work when there are other files in the file set.

So you have to create two restore jobs.

Also, a restore only creates some temporary files that can be used for a database restore.

LEVEL 4: CONFIGURATION

The file daemon configuration: `/etc/bareos/bareos-fd.d/client/myself.conf`

```
Client {
  Name = ubuntu-fd
  Plugin Directory = "/usr/lib/bareos/plugins"
  Plugin Names = "python"
}
```

LEVEL 4: CONFIGURATION

As before, we call the plugin from the `FileSet` resource:

```
FileSet {
  Name = "nextcloud-xtra"
  Description = "Backup nextcloud dirs and database with xtraback
  Include {
    Options {
      Signature = MD5 # calculate md5 checksum per file
      Compression = GZIP
    }
    File = /var/www/nextcloud
    Plugin = "python:module_path=/usr/lib/bareos/plugins:module_n
  }
}
```

LEVEL 4: CONFIGURATION

Sample script you need to call after the restore job:

```
#!/bin/sh
perconadir="/tmp/xtra-restore/_percona"
mysqldatadir="/var/lib/mysql"
mysqlservice="mysql"
mysqluser="mysql"

cd $perconadir || exit 1

backupdir=$(ls -t | head -1)

cd $backupdir || exit 1

basedir=$(ls | head -1)

inc_count=$(ls | wc -l)
inc_count=$(expr $inc_count - 1)
```

CONCLUSIONS

CONCLUSIONS

There is no one-size-fits-all solution

CONCLUSIONS

There is no one-size-fits-all solution

Assess your requirements and pick the simplest solution

CONCLUSIONS

There is no one-size-fits-all solution

Assess your requirements and pick the simplest solution

Test your restores!

CONCLUSIONS

There is no one-size-fits-all solution

Assess your requirements and pick the simplest solution

Test your restores!

REALLY, test your restores!

FINISHED!

Sebastian Lederer

dass IT GmbH

slederer@dass-it.de

9/2018

uses reveal.js/http://lab.hakim.se/reveal-js/ by Hakim El Hattab