SINE NOMINE ASSOCIATES

Flexible AFS Backup System

Ralf Brunckhorst
Sine Nomine Associates
2022 AFS Technologies Workshop

Introduction

SINE NOMINE ASSOCIATES

- Free software project to integrate OpenAFS file-servers with an existing backup scheme.
- Is a set of tools and daemons to help backup AFS cells
- Make any backup system OpenAFS compatible (without the need of a plugin)
- Layer between OpenAFS-data and backup system (e.g. Netbackup) which decouple the "get data from AFS" handling from the "store long-term backup data" handling.
- AFS data handling is more ‚flexible‘
- Dumping volume blobs to regular files on disk.
- Regular backup system handle backing up those regular files (e.g. to tape).
- Has been used in production at a large OpenAFS site for several years.

Overview

SINE NOMINE ASSOCIATES

- FABS is written in Python (python-3)
- It uses a few libraries which are easily obtainable in familiar Linux distributions as packages
- Packaging exists for RPM and Debian/Ubuntu packages
- Tested on RHEL7+8 (derivatives incl.), Debian11, Ubuntu20+22 and Fedora35+
- Installation on other Linux/Unix distributions which support python-3 should be possible (e.q. Solaris11)
- Deployed in production on RHEL7 and Debian11

Technical Details

SINE NOMINE ASSOCIATES

- Runs on a single machine per cell
- Has a single server daemon that is used for
  - orchestrating backup runs
  - checking for errors, sending reports, etc
  - intended to run under bosserver
- Runs "vos dump" and "vos restore" commands directly from that machine
  Note: Only full dump supported
- FABS jobs are scheduled and will be picked up by the “fabsys server” process
- Idempotent retriable steps
  - Operations like performing backups and restores happen in a sequence of retriable 'states'. If something fails in the middle of a backup or restore operation, the operation can be rolled back to a known-good point and retried.
  - If an operation fails too many times, we stop retrying it and instead generate an alert

Technical Details (cont.)

SINE NOMINE ASSOCIATES

- Records various volume metadata and VLDB info at the time of the backup
- Configuration of FABS is in YAML
  Edit or add new files in /etc/fabs/fabs.yaml.d/
- Configurable hooks for certain actions available which are just commands/scripts.
  Examples (Shell + Perl) included in /etc/fabs/hooks/
- No mechanism for scheduling backups, instead run FABS backup-command(s) via cron or bosserver.

Requirements

SINE NOMINE ASSOCIATES

- A krb5 keytab for authenticated access to the cell for full functionality.
  - k5start is used for those accesses
  - 'localauth'-mode possible but not recommended
- OpenAFS-client running and operational
- dumpscan
- A SQL database (supported by Python's SQLAlchemy)
  - Supported and tested: SQLite and MySQL / MariaDB
  - Default: SQLite database on local disk on the fabs server
  - Might work: Postgresql, Oracle and MS-SQL
  - Other dialects are published as external SQLAlchemy projects
- A storage directory for use as FABS blob storage
  - can have multiple directories
  - will distribute volume data across them relatively evenly
  - will failover to other storage directories if one is not functioning properly

Interface (CLI)

SINE NOMINE ASSOCIATES

- Only one main-command: **fabsys**
- Detailed man-pages for fabsys and each subcommand
- Output can also be in a machine-readable JSON format
- Subcommands available for different areas like:
  - Daemon
  - Configuration
  - Database initialization and maintenance
  - Backups
  - Dump handling
  - Restores
  - Job handling
    - Status
    - Retry
    - Kill

Workflow

SINE NOMINE ASSOCIATES

- Using FABS to dump volume blobs to path (FABS storage) on local file-system according to various configuration directives and other information (e.g. whether a volume has changed since the last dump)
  _> fabsys backup-start --all --note 'daily backup-job'
- Using a Backup system to backup the contents on file-system to tape
- Currently, the general scheme in mind for limiting space in the blob storage directory
  - is to keep only the most recent copy of a volume around
  - delete all other copies from disk (FABS storage can be trimmed by specific maintenance command)
    >_ fabsys storage-trim (examine output and delete from file-system)
  - keeping them around on tape
- In case of a restore request:
  - Find which backup of it you want to restore by volume name or path and other criteria
    >_ fabsys dump-find --path /afs/cell/user/username --near 1438491600 --admin
  - Restore the specific volume dump
    >_ fabsys restore-start --dump-id 6 -admin
  - If the needed dump is only on 'tape', FABS will run a script that can do a notification and interaction with the Backup system to get the needed dump.
  - Data will be restored to a staging location (shown by the status command)
    >_ fabsys restore-status
  - After a certain amount of time (configurable, defaults to 1 week), the staging data will be removed, and the restore request will be marked as done.

FABSYS subcommands

SINE NOMINE ASSOCIATES

```
fabsys [-h] [--version] subcommand ...

Main FABS command suite

positional arguments:
  subcommand
    vars            View fabs compile-time variables
    config          Query information about the local config
    db-clean
    db-init         Generate/execute the relevant SQL to initialize the database
    db-upgrade      Generate/execute the relevant SQL to upgrade the db from an
                    older version
    storage-init    Initialize our local storage for volume blobs
    backup-start    Manually start a new backup run
    server          Run the fabs backup server
    backup-status
                    View current backup status
    restore-status
                    View status of active restore requests
    restore-start
                    Restore a volume from backup
    dump-find
    dump-list
    dump-delete
    backup-kill     Forcibly stop a backup run from running
    restore-kill    Forcibly kill a restore request
    backup-retry    Retry a failed backup run
    restore-retry
                    Retry a failed restore request
    backup-needed
    backup-inject
    storage-trim

common options:
  -h, --help
  --version         show program's version number and exit
```

Examples: Backup

SINE NOMINE ASSOCIATES

```
>_ fabsys backup-start --volume user.2830 --note "Schedule test backup of a specific volume"
Created new backup run 207 for volume user.2830

>_ fabsys backup-status
Found 1 backup run(s)
Backup run 207 [NEW]: Schedule test backup of a specific volume
  status.0: backup run created
  cell: example.com, volume: user.2830, errors: 0
  start: Mon Jun 13 05:41:07 2022 EDT
  end:   0
  jobs: 0 total

>_ cat LOGFILE
FABS v1.0 Backup Report

Started on:  Mon Jun 13 05:41:07 2022 EDT
Finished on: Mon Jun 13 05:41:32 2022 EDT
Note: Schedule test backup of a specific volume

Attempted to backup 1 volume(s)
Successfully dumped 0 volume(s)
Failed to dump 0 volume(s)
Skipped 1 volume(s)

Volumes per partition:
  207.89.43.113 vicepa: 1 total, 0 dumped, 0 failed, 1 skipped

Volumes that were skipped because the volume was unchanged:
  user.2830 (vl_id 92072)
```

Examples: Restore

SINE NOMINE ASSOCIATES

```
>_ fabsys dump-find --volume user.2830 --near "2022-05-11" --admin
Volume dump id 818, volume: 536880651 (user.2830)
  cloned at:    Sat Jun 11 05:11:04 2022
  last updated: Sat Jun 11 05:10:39 2022
  dump blob: /var/lib/fabs/fabs-dumps/cell-example.com/20/00/26/536880651/536880651.91147.dump

>_ fabsys dump-find --path /afs/example.com/user/foo --admin
Volume dump id 815, volume: 536880118 (user.2785)
  cloned at:    Sat Jun 11 03:29:59 2022
  last updated: Fri Jun 10 09:35:05 2022
  dump blob: /var/lib/fabs/fabs-dumps/cell-example.com/20/00/23/536880118/536880118.91094.dump
Volume dump id 819, volume: 536880118 (user.2785)
  cloned at:    Mon Jun 13 03:29:24 2022
  last updated: Sun Jun 12 17:42:49 2022
  dump blob: /var/lib/fabs/fabs-dumps/cell-example.com/20/00/23/536880118/536880118.92023.dump

>_ fabsys restore-start --dump-id 818 --admin
Created new restore request id 3

>_ fabsys restore-status
Found 1 active restore request(s):

Restore request 3 [NEW]: Manually-issued restore request from command line
  status.0: restore request created
  cell: example.com, volume: user.2830 (id 536880651, voldump 818)
  errors: 0, user: [admin], requested path: [none]
  tmp dump blob: [none]
  staging volume: [none], mount: [none], time: [none]
  next update: 0, mtime: Mon Jun 13 05:59:39 2022, ctime: Mon Jun 13 05:59:39 2022
  backend request info: [none]

>_ fabsys restore-status
Found 1 active restore request(s):

Restore request 3 [RESTORE_DONE]: Manually-issued restore request from command line
  status.14: Done mounting volume
  cell: example.com, volume: user.2830 (id 536880651, voldump 818)
  errors: 0, user: [admin], requested path: [none]
  tmp dump blob: [none]
  staging volume: fabs.rreq.3, mount: /afs/.example.com/service/fabs-restore/fabs.rreq.3, time: Mon Jun 13 06:00:36 2022
  next update: 0, mtime: Mon Jun 13 06:00:36 2022, ctime: Mon Jun 13 05:59:39 2022
  backend request info: [none]
```

Example: storage trim in JSON format

SINE NOMINE ASSOCIATES

```
>_ fabsys storage-trim --format json | jq
{
  "fabs_storage_trim": [
    {
      "path": "/var/lib/fabs/fabs-dumps/cell-example.com/20/00/26/536880651/536880651.91147.dump",
      "dump": {
        "id": 818,
        "vl_id": 91147,
        "hdr_size": 58956,
        "hdr_creation": 1654938664,
        "hdr_copy": 1618837634,
        "hdr_backup": 1654938664,
        "hdr_update": 1654938639,
        "incr_timestamp": 0,
        "dump_size": 60368081,
        "dump_storid": 1,
        "dump_spath": "example.com/20/00/26/536880651/536880651.91147.dump",
        "dump_checksum": "md5:589e44964789eea1f46c439a05803815",
        "br_id": 204,
        "name": "user.2830",
        "rwid": 536880651,
        "roid": null,
        "bkid": 536880653,
        "cell": "example.com",
        "dump_blob": {
          "bstore": {
            "uuid": "4854eb57-624f-4f34-943a-649188843904",
            "storid": 1,
            "prefix": "/var/lib/fabs/fabs-dumps"
          },
          "rel_path": "example.com/20/00/26/536880651/536880651.91147.dump",
          "abs_path": "/var/lib/fabs/fabs-dumps/cell-example.com/20/00/26/536880651/536880651.91147.dump"
        }
      }
    }
  ]
}

>_ fabsys storage-trim --format json | jq -r .fabs_storage_trim[].path

/var/lib/fabs/fabs-dumps/cell-example.com/20/00/26/536880651/536880651.91147.dump
```

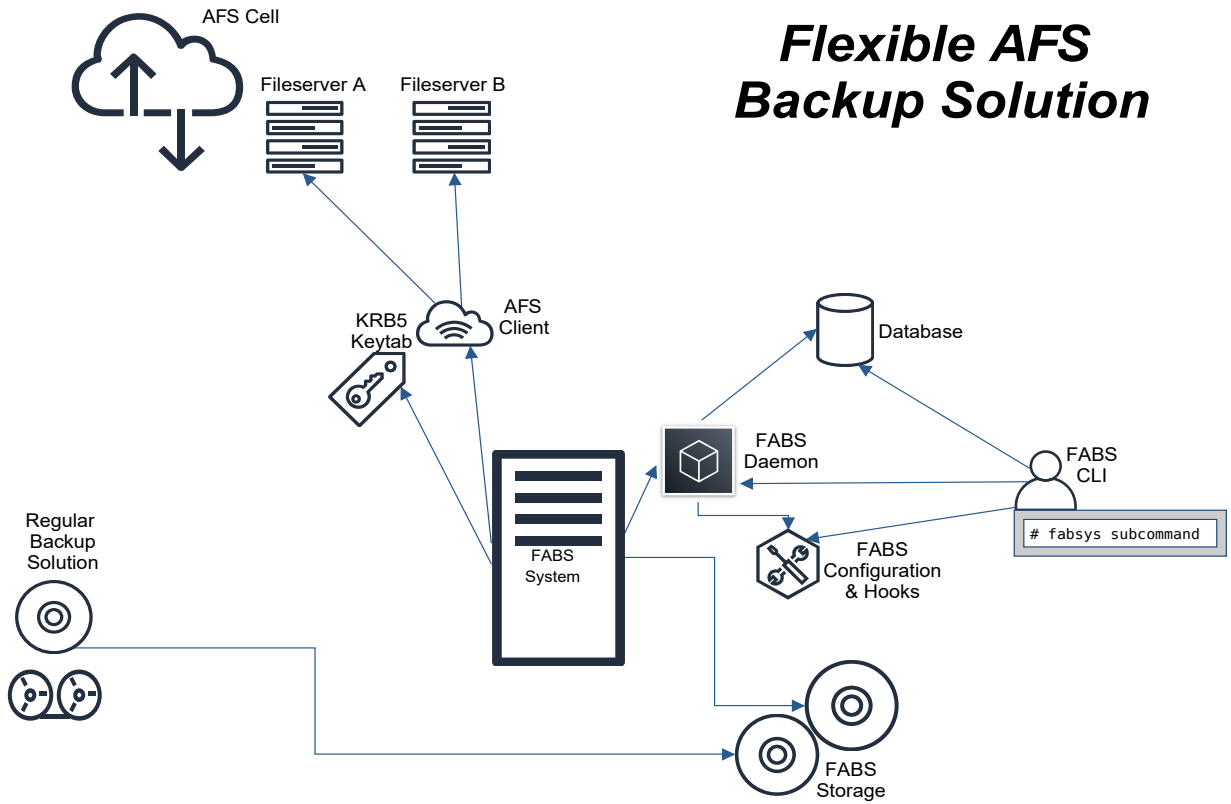
Examples: Storage & Dump maintenance

SINE NOMINE ASSOCIATES

```
# Storage-trim
>_ fabsys storage-trim
/var/lib/fabs/fabs-dumps/cell-sinenomine.net/20/00/25/536880396/536880396.4404.dump
/var/lib/fabs/fabs-dumps/cell-sinenomine.net/20/00/23/536880118/536880118.91094.dump
/var/lib/fabs/fabs-dumps/cell-sinenomine.net/20/00/21/536879511/536879511.83202.dump

# Interaction with other commands
>_ while read -r BLOB; do
     echo "Do something with BLOB-file: $BLOB"
   done < <(fabsys dump-list --redundant 1 --before "2022-06-11" --format json | \
            jq -r .fabs_dump_list.dumps[].dump_blob.abs_path)

Do something with BLOB-file: /var/lib/fabs/fabs-dumps/cell-example.com/20/00/25/536880396/536880396.4404.dump
Do something with BLOB-file: /var/lib/fabs/fabs-dumps/cell-example.com/20/00/21/536879511/536879511.83202.dump

# Delete dumps with specifc criterias
>_ fabsys dump-delete --redundant 3 --before "2022-06-11"
Deleting Volume dump id 464, volume: 536880396 (prj.test)
  cloned at:    Tue Dec 14 05:08:35 2021
  last updated: Wed Jun 26 16:47:59 2019
  dump blob: /var/lib/fabs/fabs-dumps/cell-example.com/20/00/25/536880396/536880396.4404.dump
Deleting Volume dump id 799, volume: 536879511 (user.2744)
  cloned at:    Wed May 25 03:29:20 2022
  last updated: Tue May 24 07:00:13 2022
  dump blob: /var/lib/fabs/fabs-dumps/cell-example.com/20/00/21/536879511/536879511.83202.dump
Successfully deleted 2 dumps
```

Recap

SINE NOMINE ASSOCIATES



Links

SINE NOMINE ASSOCIATES

- FABS: https://github.com/openafs-contrib/fabs
  FABS has recently become publicly available.
- dumpscan: https://github.com/openafs-contrib/cmu-dumpscan

Thank You!