Designing your Cluster Backup/Restore Process

Table of Contents

You must have a reliable, tested process for backing up your Nexus Repository Manager deployment .

This section will cover what you need to create a backup and restore process appropriate for your High Availability deployment .

High-level Overview

An appropriate backup process for Nexus Repository Manager needs to capture state from two separate, but interdependent parts of your deployment:

- the contents of each blob store
- a copy of the local storage for each node (i.e. the

  ```
  $data-dir
  ```

  directory, including the OrientDB databases within)

In the worst case scenario, a Nexus Repository Manager High Availability environment can be recreated minimally from:

- the contents of each blob store
- a complete set of OrientDB database exports from one of the nodes participating in the cluster

Backing up Shared File Systems for your Blob Store(s)

> ⓘ Nexus Repository Manager requires that you choose and execute a separate backup process appropriate for the file systems under your blob store(s).

Generic NFS

Tools that back up the file system under the blob store can be safely executed while Nexus Repository Manager is running.

NFS over Amazon Elastic File System (EFS)

If you are using file backed blob stores on top of Amazon Elastic File System, you will need to review Amazon's documentation specific to backing up EFS[1].

[1] http://docs.aws.amazon.com/efs/latest/ug/efs-backup.html

Amazon Simple Storage Service (S3)

The backup process for your S3 backed blob stores will involve creating a separate S3 "bucket" and periodically synchronizing the content from the source bucket under your blob store(s).

The AWS Command Line Interface (CLI)[2] provides an `s3 sync` command that you can invoke periodically to perform this:

- http://docs.aws.amazon.com/cli/latest/reference/s3/sync.html

There are also a number of third party tools that can perform this task.

Backing up local storage for your Nodes

Each node in your Nexus Repository Manager High Availability cluster has its own separate `$data-dir` directory. The `$data-dir /db` directory within contains OrientDB databases and requires special treatment.

The `db` directory cannot be backed up by basic file system backup tools while Nexus Repository Manager is running. Instead create an *Admin - Export databases for backup* task as described in the Backup and Restore[3] section.

Please note:

- All nodes in your Nexus Repository Manager cluster will automatically enter Read-Only mode when the backup task is running on any of the nodes
- It is recommended to schedule the backup task on at least 2 nodes. The tasks should run at different times to avoid concurrency issues. The backup from any node can be used to perform a restore, but running on more than one node ensures that there isn't a single point of failure for the backup.
- The folder specified in the backup task for a node must be available and writable by the user running Nexus Repository Manager on the nodes
- The backup task produces a set of files each time it is run that must be protected. Be sure to use a file system backup tool to store these files separately.

For the rest of the content within the `$data-dir` directory, traditional file system backup tools will suffice. It is safe to skip backing up the following directories within `$data-dir`:

-
  ```
  backup
  ```
- `cache`

[2] https://aws.amazon.com/cli/

[3] https://help.sonatype.com/display/NXRM3/Backup+and+Restore

- `db`
  (backed up via the *Admin - Export databases for backup* task)
- `elasticsearch` (generated via the *Repair - Rebuild respository search* task)
- `logs` (see the Operating your cluster[4] section for preserving logs)
- `tmp`

> ⓘ We recommend you execute backup of your local storage at the same starting time as your blob store file system backup.

Restoring from backup

> If you determine you need to restore your Nexus Repository Manager deployment from backup, you must first focus only on starting one node and stabilizing it before you add new nodes to the High Availability cluster.
>
> Start this process by ensuring all Nexus Repository Manager nodes in your cluster are offline.

Assuming all nodes in your cluster are offline:

1. Identify the host that will be the first node of your restored deployment. Locate the Orient DB exports from that node and identify the date and time they were taken. Use this same (or as close to) date and time when restoring the blob store file systems.
2. Restore the blob store file systems and host connectivity. Be sure to restore the blob store file system to the exact same absolute file system path used previously; don't try to restore to a different file system path.
3. Restore the local storage (`$data-dir`) for the node. Again, preserve the same absolute file system paths used previously.
4. Remove the following directories from `$data-dir/db`
   - `component`
   - `config`
   - `security`
5. Copy the complete set of OrientDB exports from that node to `$data-dir/restore-from-backup` for restoration (**Note**: For version 3.10.0 or earlier use `$data-dir/backup` as the restore location).
6. Start Nexus Repository Manager on only this node.

[4] https://help.sonatype.com/display/NXRM3/Operating+your+cluster

Confirm that your instance is in a stable state before proceeding:

- Inspect configuration and confirm it matches expectations
- Attempt to retrieve and/or publish new components

Once the first node is online and stable, it is safe to proceed in bringing additional nodes back into your cluster. Focus on adding only one node at a time to the cluster. On each of the additional nodes, delete the following folders from within `$data-dir` `/db` prior to starting Nexus Repository Manager

- `component`
- `config`
- `security`

The OrientDB databases removed from the additional node will automatically be rebuilt from the instances already running in the High Availability cluster.