General: How do I upgrade to Bright 8.0 ?

How do I upgrade from Bright 6.0/6.1/7.0/7.1/7.2/7.3 to Bright 8.0?

The procedure below can be used to upgrade a Bright 6.0 or 6.1 or 7.0 or 7.1 or 7.2 or 7.3 installation, including those with failover, to Bright 8.0

Supported Linux distributions

An upgrade to **Bright 8.0** is supported for Bright **6.0**, **6.1, 7.0, 7.1, 7.2** or **7.3** clusters that are running the following Linux distributions:

- RedHat Enterprise Linux 6.7 (RHEL6u7)
- RedHat Enterprise Linux 6.8 (RHEL6u8)
- RedHat Enterprise Linux 6.9 (RHEL6u9)
- RedHat Enterprise Linux 7.2 (RHEL7u2)
- RedHat Enterprise Linux 7.3 (RHEL7u3)
- RedHat Enterprise Linux 7.4 (RHEL7u4)
- CentOS Linux 6.7 (CENTOS6u7)
- CentOS Linux 6.8 (CENTOS6u8)
- CentOS Linux 6.9 (CENTOS6u9)
- CentOS Linux 7.2 (CENTOS7u2)
- CentOS Linux 7.3 (CENTOS7u3)
- CentOS Linux 7.4 (CENTOS7u4)
- Scientific Linux 6.7 (SL6u7)
- Scientific Linux 6.8 (SL6u8)
- Scientific Linux 6.9 (SL6u9)
- Scientific Linux 7.2 (SL7u2)
- Scientific Linux 7.3 (SL7u3)
- Scientific Linux 7.4 (SL7u4)
- SuSE Linux Enterprise Server 11 Service Pack 3 (SLES11sp3)
- SuSE Linux Enterprise Server 11 Service Pack 4 (SLES11sp4)
- SuSE Linux Enterprise Server 12 Service Pack 1 (SLES12sp1)
- SuSE Linux Enterprise Server 12 Service Pack 2 (SLES12sp2)
- SuSE Linux Enterprise Server 12 Service Pack 3 (SLES12sp3)

Prerequisites

- Extra base distribution RPMs will be installed by yum/zypper in order to resolve dependencies that might arise as a result of the upgrade. Hence the base distribution repositories must be reachable. This means that the clusters that run the Enterprise Linux distributions (RHEL and SLES11) must be subscribed to the appropriate software channels.
- Packages in /cm/shared are upgraded, but the administrator should be aware of the following:
  - If /cm/shared is installed in the local partition, then the packages are upgraded. This may not be desirable for users that wish to retain the old behavior.
  - If /cm/shared is mounted from a separate partition, then unmounting it will prevent upgrades to the mounted partition, but will allow new packages to be installed in /cm/shared within the local partition. This may be desirable for the administrator, who can later copy over updates from the local /cm/shared to the remote /cm/shared manually according to site specific requirements.
    Since unmounting of mounted /cm/shared is carried out by default, a local /cm/shared will have files from any packages installed there upgraded. According to the yum database, the system is then upgraded even though the files are misplaced in the local partition. However, the newer packages can only be expected to work properly if their associated files are copied over from the local partition to the remote partition
  - If the /cm/shared will be unmounted during the upgrade (i,e if an in-place upgrade is not being performed), then please make sure that the contents of the local /cm/shared are in sync with the remote copy.
- **IMPORTANT**:
  - Hadoop deployments must be removed (using cm-hadoop-setup), before proceeding with the upgrade. Please contact Bright Support for further assistance.
  - Bright OpenStack deployments must be removed (using cm-openstack-setup). All older Bright OpenStack packages and dependencies must be removed prior to starting the upgrade. Please contact Bright Support for further assistance.

Important note about package upgrades

The upgrade process will not only upgrade CMDaemon and its dependencies, but it will also upgrade other packages. This means that old packages will not be available from the repositories of the latest version of Bright (in this case 8.0 repositories). In some cases, this will require recompiling the user applications to use the upgraded versions of the compilers and the libraries. Also, the configurations of the old packages will not be copied automatically to the new packages, which means that the administrator will have to adjust the configuration from the old packages to suit the new packages manually.

Important note about monitoring *data* and *configuration*

The monitoring backend in Bright 8.0 has changed considerably and hence it is not possible to migrate older monitoring configuration and data to the new monitoring system. As a result, the following are to be expected after upgrading a cluster to Bright 8.0.

Monitoring data: All monitoring data from prior to the upgrade are lost after the upgrade to Bright 8.0.

Monitoring configuration: Monitoring configuration is reset to a default Bright 8.0 similar to what is configured on a freshly installed Bright 8.0 cluster. This means that all old custom monitoring configurations are lost.

Important note about GPU integration

Bright 8.0 has switched to **Nvidia DCGM** for managing and monitoring Nvidia GPUs. Only **Tesla** GPUs (**K80** and later) are supported by DCGM. However, after upgrading to Bright 8.0, it is still possible to use Bright to obtain metrics from older GPUs by following the article How to collect metrics from older GPUs using NVML, but configuring these GPUs from Bright is no longer possible.

Enable the upgrade repo and install the upgrade RPM

Install the Bright Cluster Manager upgrade RPM on the Bright head node(s) as shown below:

1. Add and enable the upgrade repo

Create a repo file with the following contents:

```
[cm-upgrade-80]

name=Bright 8.0 Upgrade Repository

baseurl=http://support.brightcomputing.com/upgrade/8.0/<DIST>/updates

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-cm
```

Note: Plese replace <DIST> with one of : rhel/7 , rhel/6, sles/12, sles/11
On RHEL based distributions, save file to `/etc/yum.repos.d/`

On SLES based distributions, save file to `/etc/zypp/repos.d/`

2. Install RPM

```
yum install cm-upgrade-8.0
```

3. Make `the cm-upgrade` command available in the default PATH

```
module load cm-upgrade/8.0
```

General: How do I upgrade to Bright 8.0 ?

Upgrade procedure

The recommended order for upgrade is:

1. Power off regular nodes.
   Terminate cloud nodes and cloud directors

2. Apply existing updates to Bright 6.0/6.1/7.0/7.1/7.2/7.3 on head node and in the software images.

   - Update head node:
     RHEL derivatives:
     ```
     yum update
     ```
     SLES derivatives:
     ```
     zypper up
     ```
   - Update software images. For each software image run the :
     RHEL derivatives:
     ```
     yum --installroot /cm/images/<software image> update
     ```
     SLES derivatives:
     ```
     zypper --root /cm/images/<software image> up
     ```

     Note: If the software image repositories differ from the repositories that the head node uses, then you should chroot into the software image first before attempting to run "yum update" or "zypper up" . This is because using the --installroot or --root switch will not allow yum/zypper to use the repositories defined in the software images.

3. Upgrade head nodes to Bright 8.0:
   ```
   cm-upgrade -u 8.0
   ```
   Important: this must be run on ***both head nodes*** *in a high availability setup*
   Recommended: Upgrade ***active head node*** *first and then the* ***passive head node***

4. Run post upgrade actions (*must be run only on the* ***active head node***):
   ```
   cm-upgrade -u 8.0 -f -p
   ```

5. In a HA setup, after upgrading both the head nodes, resync the databases. Run the following from the **active** head node (it is very important to complete this step before moving to the next one):
   ```
   cmha dbreclone <secondary>
   ```

6. Upgrade the software image(s) to Bright 8.0
   ```
   cm-upgrade -u 8.0 -i all
   ```
   Important: this must be run only on the ***active head node.*** *If the software images are not under the standard location, which is* ***/cm/images/*** *on the head node, then the option*

*"**-a**" should be used "**cm-upgrade -u 8.0 -a /apps/images -i <name of software image>**"*

7. Power on the regular nodes, cloud nodes and cloud directors

Usage and help

For more detailed information on usage, examples, and a full description:

- `cm-upgrade`
  without any arguments prints the usage and several examples on how to use the script.
- `cm-upgrade --help`
  prints the complete help and description

Upgrading using a Bright DVD/ISO

When using a Bright DVD/ISO to perform the upgrade, it is important to use a DVD/ISO that is not older than 8.0-4. The DVD/ISO version can be found (assuming that the DVD/ISO is mounted under /mnt/cdrom) with a find command such as:

```
# find /mnt/cdrom -type d -name '8.0-*'


/mnt/cdrom/data/packages/8.0-5
```

FAQs and Troubleshooting

Q: Why are my SGE or Torque jobs not running after upgrading to Bright 8.0 ?

A: This is mostly because there is an obsolete broken prolog symlink

/cm/local/apps/sge/var/prologs/10-prolog-healthchecker

or

/cm/local/apps/torque/var/prologs/10-prolog-healthchecker

Solution: Remove the broken symlink on the nodes and re-submit job.

Q: Why is the Bright package `perl-Config-IniFiles` on my SLES11 cluster not upgraded to 8.0 ?

A: This will happen when `zypper` cannot find a dependency package `perl-List-MoreUtils` and will skip

updating `perl-Config-IniFiles`

Solution: Enable a repository that contains the perl-List-MoreUtils rpm and then run:

```
zypper update perl-Config-IniFiles
```

Q: Why did `cm-upgrade` fail at the stage: 'Installing distribution packages' or 'Upgrading packages to Bright 8.0' ?

A: This will happen when some distribution package dependencies could not be met. Please look in

`/var/log/cm-upgrade.log` for detailed information about what packages are missing.

Solution: Enable required additional base distribution repositories and re-run *cm-upgrade* with the *-f* option.

Example: `cm-upgrade -u 8.0 -f`

Q: After upgrading from Bright 6.0 to Bright 8.0, why is the MySQL healthcheck failing because the cmdaemon monitoring database engine is not MyISAM ?

A: This is because Bright versions before 6.1 use InnoDB as the MySQL engine. Starting with Bright 6.1, MyISAM is the default monitoring database engine.

Solution: Change the engine type for the cmdaemon_mon database to MyISAM.

Q: Why are LDAP users sometimes not accessible on SLES compute node after upgrading to Bright 8.0 ?

A: This is most likely because the 'sssd' service failed to start. This can happen when `/var/lib/sssd` is in the exclude lists of the node or category.

Solution: Remove `/var/lib/sssd` from the exclude lists and then reboot the nodes.

Q: Why is the mvapich package not upgraded to the Bright 8.0 version ?

A: This is because support for the mvapich package has been dropped in Bright 8.0. The package is not obsoleted or removed automatically, because there might be user applications that are still using them.

Solution: If there are no user applications that use mvapich, then the package must be manually removed by the administrator.

Unique solution ID: #1360
Author: Johnny Devaprasad
Last update: 2018-02-19 12:50