Introduction

This document describes the procedure for upgrading Bright clusters to version 9.0. It should be read very carefully before carrying out an upgrade. The Bright support team can be contacted (https://www.brightcomputing.com/support) if there are Bright upgrade issues for which the documentation is insufficient.

Supported Bright Versions and Linux Distributions

Upgrades from the following Bright versions are supported:

- Bright 8.2
- Bright 8.1
- Bright 8.0
- Bright 7.3
- Bright 7.2
- Bright 7.1

The following Linux distributions are supported:

- RedHat Enterprise Linux 7 (update 7.6 and higher)
- CentOS Linux 7 (update 7.6 and higher)
- SuSE Linux Enterprise Server 12 SP3 (SLES12 SP3) and higher
- SuSE Linux Enterprise Server 15 SP1 (SLES15 SP1) and higher

Note: Upgrading RedHat 8, CentOS 8, and Ubuntu clusters are not supported at the moment, but will be added in a future update.

Important notes about upgrades from SLES12 SP3

If running SLES12 SP3, then the Bright upgrade procedure must be carried out first. Distro upgrade to SLES12 SP4 must be performed only after the Bright upgrade has been completed.

Important note about workload managers

Workload manager configurations must be disabled (using wlm-setup) before proceeding with the upgrade. All workload managers distributed by Bright (**Slurm**, **PBS Pro**) will be upgraded to the latest available versions.

Slurm configuration in 9.0 relies on **configuration overlays** by default. The roles from older Bright versions are unlikely to fit into the new scheme. The new roles placement within configuration overlays must therefore be planned ahead by the administrator, or must be adjusted after the upgrade. A sensible approach would be to make a note of the original role assignments, then adjust the working default configuration overlays step by step, checking after each step that each adjustment works as expected.

Torque and **SGE** workload managers are not supported in Bright 9.0, and hence the relevant packages will be removed as part of the upgrade.

After the upgrade has been completed, workload management must be re-deployed using the tool cm-wlm-setup.

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 9.0 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

This section will apply when support for upgrades from Bright versions prior to 8.0 become available. Upgrades from 8.0/8.1/8.2 to 9.0 are not affected.

The following are to be expected after upgrading a cluster that is running Bright versions 7.3 and lower:

The monitoring backend has changed considerably starting from Bright 8.0. Hence it is not possible to migrate the older monitoring configuration and data to the new monitoring system.

Monitoring data: All monitoring data from prior to the upgrade are lost after the upgrade to Bright 9.0

Monitoring configuration: The monitoring configuration is reset to a default Bright 9.0 configuration, similar to what is configured on a freshly-installed Bright 9.0 cluster. This means that all old custom monitoring configurations will be lost.

Important note about GPU integration

Starting from Bright 8.0, Nvidia DCGM is used for managing and monitoring Nvidia GPUs. After upgrading to Bright 9.0, 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 (https://community.brightcomputing.com/question/5d6614ba08e8e81e885f19c3); however configuring these GPUs from Bright is no longer possible.

Other 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.

- **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.

- **Kubernetes** deployments from older Bright versions must be removed before upgrading and re-deployed after upgrade.

- Configurations of **cluster extension** to the cloud must be removed before upgrading and re-deployed after upgrade.

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 **9.0-6**. 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 '9.0-*'
```

```
/mnt/cdrom/data/packages/9.0-6
```

Enable upgrade

Enable the upgrade repo and install package

- RHEL derivatives:

  ```
  $ yum-config-manager \
        --add-repo http://support.brightcomputing.com/upgrade/9.0/rhel/7/updates
  $ yum --nogpgcheck install cm-upgrade-9.0
  ```

- SLES12:

  ```
  $ zypper addrepo \
        http://support.brightcomputing.com/upgrade/9.0/sles/12/updates cm-upgrade-9.0
  $ zypper install cm-upgrade-9.0
  ```

- SLES15:

```
$ zypper addrepo \
      http://support.brightcomputing.com/upgrade/9.0/sles/15/updates cm-upgrade-9.0
$ zypper install cm-upgrade-9.0
```

Load the environment module

```
$ module load cm-upgrade/9.0
```

Perform upgrade

Power off nodes

- Power off regular nodes

- Terminate cloud nodes and cloud directors

Apply updates to head node

- RHEL derivatives:

  ```
  $ yum update
  ```

- SLES derivatives:

  Important: Must not be done for clusters running SLES12 SP3, if the following command will result in an upgrade to SLES12 SP4. Upgrade to SLES12 SP4 must be carried out only after the Bright upgrade procedure has been completed.

  ```
  $ zypper up
  ```

Apply updates to software images

For each software image, do the following:

- RHEL derivatives:

  ```
  $ yum --installroot /cm/images/<software image> update
  ```

- SLES derivatives:

  Important: Must not be done for clusters running SLES12 SP3, if the following command will result in an upgrade to SLES12 SP4. Upgrade to SLES12 SP4 must be carried out only after the Bright upgrade procedure has been completed.

  ```
  $ 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.

**Upgrade head nodes to Bright 9.0**

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.

- Upgrade using repositories accessible over the network

  ```
  $ cm-upgrade
  ```

- Upgrade using a Bright DVD/ISO

  ```
  $ cm-upgrade -b /root/bright9.0-centos7u7.iso
  ```

In an 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>
```

Reboot head node(s)

The head node(s) must be rebooted before proceeding to running the post upgrade actions.

Post upgrade head node

Important: this must be run on both head nodes in a high availability setup

```
$ module load cm-upgrade/9.0
```

```
$ cm-post-upgrade -m
```

Upgrade the software image(s) to Bright 9.0

Important: this must be run only on the active head node.

- Upgrade using repositories accessible over the network

  ```
  $ cm-upgrade -i all
  ```

- Upgrade using a Bright DVD/ISO

```
$ cm-upgrade -i all -b /root/bright9.0-centos7u7.iso
```

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.

```
Examples:
$ cm-upgrade -a /apps/images -i <name of software image>
$ cm-upgrade -a /apps/images -i <name of software image> -b /root/bright9.0-centos7u7.iso
```

Post upgrade software images

Important: this must be run only on the active head node.

```
$ cm-post-upgrade -i all
```

Upgrade /cm/node-installer to Bright 9.0

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

- Upgrade using repositories accessible over the network

  ```
  $ cm-upgrade -x
  ```
- Upgrade using a Bright DVD/ISO

  ```
  $ cm-upgrade -x -b /root/bright9.0-centos7u7.iso
  ```