NetBrain® Integrated Edition 8.0

System Upgrade Guide

Two-Server Deployment

Version 8.03 | Last Updated 2020-12-08

Contents

1. Upgrading System 3
   1.1. Terminating System Tasks 4
   1.2. Stopping Server Services 4
   1.3. Backing Up MongoDB Data 5
   1.4. Upgrading NetBrain Database Server 6
   1.5. Upgrading NetBrain Application Server 10
   1.6. Verifying Upgrade Results 14
   1.7. Customizing MongoDB Disk Alert Rules 16
   1.8. Tuning Live Access 17
   1.9. Scheduling Benchmark Task 17
2. Appendix: Offline Installing Third-party Dependencies 18
3. Appendix: Editing a File with VI Editor 19
4. Appendix: Restoring MongoDB Data 20
5. Appendix: Dumping MongoDB Data 22

1. Upgrading System

The upgrade process ensures data integrity, which means that the data in the current system will be still available after upgrading. If you encounter any issues during the upgrade process, contact NetBrain Support Team for help.

> **Note:** Before upgrading your system, check its current version and the network connectivity requirements.

Upgrade from IEv8.0x

1. Terminate System Tasks
2. Stop Server Services
3. Back Up MongoDB Data
4. Upgrade NetBrain Database Server
5. Upgrade NetBrain Application Server
6. Verify Upgrade Results
7. Customize MongoDB Disk Alert Rules
8. Tune Live Access
9. Schedule Benchmark Task

> **Note**: If you have installed Smart CLI and Ansible Agent before, you need to run the latest installation packages to upgrade the two components. See Installing Smart CLI and Installing Ansible Agent for more details.
>
> To obtain the installation package of Ansible Agent:
>
> - **Option 1:** If the Linux server has no access to the Internet, obtain the **netbrain-ansibleagent-linux-x86_64-rhel7-8.0.3.tar.gz** file from NetBrain and then upload it to the **/opt/netbraintemp8.0.3** directory by using a file transfer tool.
> - **Option 2:** If the Linux server has access to the Internet, run the `wget http://download.netbraintech.com/netbrain-ansibleagent-linux-x86_64-rhel7-8.0.3.tar.gz` command under the **/opt/netbraintemp8.0.3** directory to directly download the **netbrain-ansibleagent-linux-x86_64-rhel7-8.0.3.tar.gz** file from NetBrain official download site.

Network Connectivity Requirements

Source	Destination	Protocol and Port Number *)
Thin Client Service Monitor Agent	Application Server	HTTP/HTTPS (80/443)

Source	Destination	Protocol and Port Number *)
Application Server	Knowledge Cloud Domain (https://knowledgecloud.netbraintech.com/)	HTTPS (443)
Application Server	Database Server	TCP 5672/9200/27017/27654 TCP 6379 (non-SSL)/7000 (SSL)
Application Server	Ansible Agent (add-on)	TCP 50051
Application Server	Live Network	ICMP/SNMP/Telnet/SSH/REST API

> **Note:** *) The port numbers listed in this column are defaults only. The actual port numbers used during installation might be different.

1.1. Terminating System Tasks

1. Log in to the System Management page.
2. In the System Management page, select **Operations > Task Manager** from the quick access toolbar.
3. Select all running tasks and click **End Process**.

1.2. Stopping Server Services

To avoid any further dataset changes or data corruption while reinstalling MongoDB/Elasticsearch binary files or restoring MongoDB/Elasticsearch data, stop the relevant services.

Log in to the Windows server where NetBrain applications are installed and stop the following services in the Task Manager.

NetBrain Component	Service Name
Web API Server	W3SVC
Web API Server	WAS
Worker Server	NetBrainWorkerServer
Front Server	NetBrainFrontServer
Task Engine	NetBrainTaskEngine
Front Server Controller	NetBrainFrontServerController

Service Monitor Agent	NetBrain Agent
Knowledge Cloud Proxy	NetBrainKCProxy

1.3. Backing Up MongoDB Data

Before upgrading the system, it is highly recommended to back up all MongoDB data in case of any data loss or corruption during the upgrade process. The backup data will be used to restore data after MongoDB is reinstalled. See Restoring MongoDB Data for more details.

In case that you don't want to stop the service of MongoDB or the volume of the MongoDB data is small, you can refer to Dumping MongoDB Data for another way to back up the data and refer to Restoring Dumped MongoDB Data to restore the dumped data.

The following steps introduces how to use the `cp` command to copy underlying MongoDB data files directly for backup.

> **Notes:**
>
> – Make sure you have stopped all relevant services before backing up data.
>
> – The backup data can only be used to restore the database on the same server.

1. Log in to the Linux server where the MongoDB node is installed as the **root** user.
2. Stop the service of MongoDB.
   1) Run the `systemctl stop mongod` command to stop the MongoDB service.
   2) Run the `ps -ef|grep mongod` command to verify whether the **mongod** process is stopped.

```
[root@localhost ~]# ps -ef| grep mongod
root     15136 14237  0 10:42 pts/2    00:00:00 grep --color=auto mongod
```

> **Note:** If the **mongod** process is stopped, the result should only contain one entry as shown above.

3. Run the `mkdir /etc/mongodb_databk` command to create a directory under the **/etc** directory to save the backup data.

```
[root@localhost ~]# mkdir /etc/mongodb_databk
```

> **Note:** Ensure the backup directory (**/etc/mongodb_databk** in this example) has sufficient space to store the backup data.

4. Run the `cd /usr/lib/mongodb` command to navigate to the **/usr/lib/mongodb** directory.

> **Note:** If you modified the following default directory to store all MongoDB data files during the MongoDB installation, you must use the new directory (available in **mongod.conf**) accordingly.

5. Run the `du -hs data` command under the **/usr/lib/mongodb** directory to check the total size of MongoDB backup data.
6. Run the `cp -a data /etc/mongodb_databk` command under the **/usr/lib/mongodb** directory to copy all MongoDB data files from the **data** directory to the **/etc/mongodb_databk** directory.

```
[root@localhost mongodb]# cp -a data /etc/mongodb_databk
```

7. Run the `cd /etc/mongodb_databk` command to navigate to the **/etc/mongodb_databk** directory.
8. Run the `ls -al` command under the **/etc/mongodb_databk** directory to browse the backup data.

```
[root@localhost mongodb_databk]# ls -al
total 136
drwxr-xr-x.  3 root root         18 Jun 6 22:49 .
drwxr-xr-x.  6 root root         79 Jun 6 22:48 ..
drwxr-xr-x.  4 netbrain netbrain 106496 Jun 6 22:49 data
```

9. Run the `systemctl start mongod` command to start the MongoDB service.

1.4. Upgrading NetBrain Database Server

Pre-Upgrade Tasks

RabbitMQ has dependencies on the third-party packages **socat** and **logrotate.** Run the `rpm -qa|grep socat` and `rpm -qa|grep logrotate` commands to check whether **socat** and **logrotate** have been installed on this Linux server. If they have not been installed yet, you can choose either option below to install the dependencies:

- **Online Install:** run the `yum -y install socat` and `yum -y install logrotate` commands to install them online.
- **Offline Install:** refer to Appendix: Offline Installing Third-party Dependencies for further instructions.

- Service Monitor Agent has dependencies on the third-party package **libffi-devel zlib-devel readline-devel bzip2-devel ncurses-devel gdbm-devel xz-devel tk-devel libffi-devel**. Run the `rpm -qa|grep -E "zlib-devel|readline-devel|bzip2-devel|ncurses-devel|gdbm-devel|xz-devel|tk-devel|libffi-devel"` command to check whether it has been installed on this Linux server. If it has not been installed yet, you can choose either option below to install the dependencies:
  - **Online Install:** run the `yum -y install zlib-devel readline-devel bzip2-devel ncurses-devel gdbm-devel xz-devel tk-devel libffi-devel` command to install it online
  - **Offline Install:** refer to Appendix: Offline Installing Third-party Dependencies for further instructions.

Upgrading NetBrain Database Server

1. Log in to the Linux server as the **root** user.
2. Run the `mkdir` command to create a directory under the **/opt** directory to place the installation package. For example, **netbraintemp8.0.3**.

   > **Note**: Don't place the installation package under any personal directories, such as **/root**.

3. Download the installation package.
   - **Option 1:** If the Linux server has no access to the Internet, upload the **netbrain-all-in-two-linux-x86_64-rhel7-8.0.3.tar.gz** file to the **/opt/netbraintemp8.0.3** directory by using a file transfer tool.
   - **Option 2:** If the Linux server has access to the Internet and the **yum** service has been installed:
     1) Run the `yum -y install wget` command to install the `wget` command if it has not been installed.
     2) Run the `cd /opt/netbraintemp` command to navigate to the **/opt/netbraintemp8.0.3** directory.
     3) Run the `wget http://download.netbraintech.com/netbrain-all-in-two-linux-x86_64-rhel7-8.0.3.tar.gz` command under the **/opt/netbraintemp8.0.3** directory to download the **netbrain-all-in-two-linux-x86_64-rhel7-8.0.3.tar.gz** file to this directory:

        > **Note:** The download link is case-sensitive.

4. Run the `tar -zxvf netbrain-all-in-two-linux-x86_64-rhel7-8.0.3.tar.gz` command under the **/opt/netbraintemp8.0.3** directory to extract installation files.

```
[root@localhost netbraintemp8.0.3]# tar -zxvf netbrain-all-in-two-linux-x86_64-rhel7-
8.0.3.tar.gz
netbrain-all-in-two-linux-8.0.3/
netbrain-all-in-two-linux-8.0.3/config/
...
netbrain-all-in-two-linux-8.0.3/upgrade.sh
```

5. Run the `cd netbrain-all-in-two-linux-8.0.3` command to navigate to the **netbrain-all-in-two-linux-8.0.3** directory.
6. Run the `systemctl start mongod` command to restart the MongoDB service.
7. Run the `./upgrade.sh` command under the **netbrain-all-in-two-linux-8.0.3** directory to install NetBrain Linux components.
   1) Read the license agreement, and then type **YES** and press the **Enter** key.
   2) Type **I ACCEPT** and press the **Enter** key to accept the license agreement. The script starts to check whether the system configuration of the Linux server meets the requirement, and all required dependent packages are installed for each Linux component.

```
[root@localhost netbrain-all-in-two-linux-8.0.3]# ./upgrade.sh

Please read the End User License Agreement ("EULA") for the license type (perpetual or
subscription)
purchased in the order form at https://www.netbraintech.com/legal-tc/ carefully. I have read
the
subscription EULA, if I have purchased a subscription license, or the perpetual EULA, if I
have
purchased a perpetual license, at the link provided above. Please type "YES" if you have read
the
applicable EULA and understand its contents, or "NO" if you have not read the
applicable EULA. [YES/NO]: YES

Do you accept the terms in the subscription EULA, if you have purchased a subscription
license, or
the perpetual EULA, if you have purchased a perpetual license? If you accept, and to continue
with
the installation, please type "I ACCEPT" to continue. If you do not accept, and to quit the
installation script, please type "CANCEL" to stop. [I ACCEPT/CANCEL]: I ACCEPT
INFO: 2020-07-21 21-33-33.561: Creating upgrading log...
INFO: 2020-07-21 21-33-33.610: Starting to checking system
INFO: 2020-07-21 21-33-38.514: Collecting system information SUCCEEDED.
INFO: 2020-07-21 21-33-38.524: Starting to check Linux OS info
INFO: 2020-07-21 21-33-38.549: ALL_COMPONENT_LIST=mongodb licenseagent elasticsearch rabbitmq
redis servicemonitoragent
```

8. MongoDB, License Agent, Elasticsearch, RabbitMQ, Redis and Service Monitor Agent will be upgraded sequentially. Then run the `reboot` command to restart the machine.

9. Run the following commands to check their service status separately after the server starts up.

   - **`systemctl status mongod`**

```
[root@localhost ~]# systemctl status mongod
  mongod.service - MongoDB service
  Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
  Active: activing (start) since Tue 2020-07-07 13:23:24 EDT; 5min ago
  Control: 1143 (mongod)
  Memory: 902.3M (limit: 8.0G)
...
```

   - **`systemctl status netbrainlicense`**

```
[root@localhost ~]# systemctl status netbrainlicense
  netbrainlicense.service - NetBrain license agent service
  Loaded: loaded (/usr/lib/systemd/system/netbrainlicense.service; enabled; vendor preset:
disabled)
  Active: active (running) since Tue 2020-07-07 13:23:35 EDT; 9min ago
  Process: 1175 ExecStart=/usr/bin/netbrainlicense/licensed -f
/etc/netbrainlicense/licensed.conf (code=exited, status=0/SUCCESS)
  Process: 1150 ExecStartPre=/bin/chmod o+r /sys/class/dmi/id/product_uuid (code=exited,
status=0/SUCCESS)
 Main PID: 2111 (licensed)
   Memory: 17.6M
...
```

- **`systemctl status elasticsearch`**

```
[root@localhost ~]# systemctl status elasticsearch
  elasticsearch.service - Elasticsearch
  Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset:
disabled)
  Active: active (running) since Tue 2020-07-07 13:23:25 EDT; 12min ago
  Docs: http://www.elastic.co
  Main PID: 1153 (java)
  Memory: 4.5G
...
```

- **`systemctl status rabbitmq-server`**

```
[root@localhost ~]# systemctl status rabbitmq-server
  rabbitmq-server.service - RabbitMQ broker
  Loaded: loaded (/usr/lib/systemd/system/rabbitmq-server.service; enabled; vendor preset:
disabled)
  Active: active (running) since Tue 2020-07-07 13:26:56 EDT; 9min ago
  Main PID: 1136 (beam.smp)
  Status: "Initialized"
  Memory: 131.2M
...
```

- **`systemctl status redis`**

```
[root@localhost ~]# systemctl status redis
  redis.service - Redis
  Loaded: loaded (/usr/lib/systemd/system/redis.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2020-07-07 13:23:25 EDT; 14min ago
 Main PID: 1159 (redis-server)
   Memory: 7.3M
...
```

- **`systemctl status netbrainagent`**

```
[root@localhost ~]# systemctl status netbrainagent
  netbrainagent.service - NetBrain Service Monitor Agent Daemon
  Loaded: loaded (/usr/lib/systemd/system/netbrainagent.service; enabled; vendor preset:
disabled)
  Active: active (running) since Tue 2020-07-07 13:23:24 EDT; 15min ago
  Main PID: 1144 (python3)
  Memory: 135.0M
...
```

10. Verify the upgraded MongoDB version.

    1) Log in to the Linux server where MongoDB is installed.

    2) Open a command prompt and run the `mongo --host <IP or hostname of MongoDB Server:Port> -u <username> -p <password> --authenticationDatabase <database_name> --authenticationMechanism SCRAM-SHA-256` command to connect to MongoDB.

       Example:

```
[root@localhost ~]# mongo --host 10.10.3.142:27017 -u mongodb -p mongodb --
authenticationDatabase admin --authenticationMechanism SCRAM-SHA-256
MongoDB shell version v4.0.6
```

```
connecting to: mongodb://10.10.3.142:27017/?authMechanism=SCRAM-SHA-
256&authSource=admin&gssapiServiceName=mongodb
...
```

> **Tip:** If SSL is enabled, run the `mongo --host <IP or hostname of MongoDB Server:Port> -u <username> -p <password> --authenticationDatabase <database_name> --ssl --sslAllowInvalidCertificates --authenticationMechanism SCRAM-SHA-256` command.

1.5. Upgrading NetBrain Application Server

Pre-Upgrade Tasks

> **Note:** Before the upgrading, clean the **C:\Windows\Temp** folder to make sure the upgrade process goes smoothly.

Complete the following steps to back up the configuration files for your Front Server and Front Server Controller before upgrading the Application Server.

1. Navigate to the **C:\Program Files\NetBrain\Front Server\conf** directory, and then back up all files WITHOUT the prefix "**fix**_".
2. Navigate to the **C:\Program Files\NetBrain\Front Server Controller\conf** directory, and then back up all files WITHOUT the prefix "**fix**_".

> **Note: C:\Program Files\NetBrain** is the default installation directory of the Application Server. If you changed it during the Application Server installation, modify it accordingly.

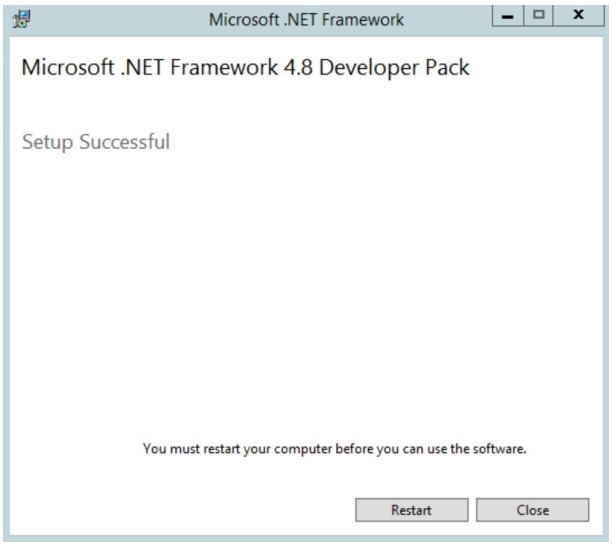
Upgrading NetBrain Application Server

Complete the following steps with administrative privileges.

1. Download the **netbrain-all-in-two-windows-x86_64-8.0.3.zip** file from http://download.netbraintech.com/netbrain-all-in-two-windows-x86_64-8.0.3.zip and save it in your local folder.
2. Extract files from the **netbrain-all-in-two-windows-x86_64-8.0.3.zip** file.
3. Navigate to the **netbrain-all-in-two-windows-x86_64-8.0.3** folder, right-click the **netbrain-application-8.0.3.exe** file and then select **Run as administrator** to launch the Installation Wizard.
4. Follow the installation wizard to complete the installation step by step:

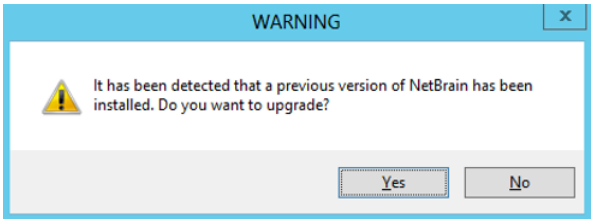
1) If **.NET Framework 4.8** has not been pre-installed on this machine, the Installation Wizard will guide you through the installation of **.NET Framework 4.8**.

> **Note:** Make sure the Windows update is of the latest. For Windows Server 2012, the update **KB2919442** and **KB2919355** must be installed before the .NET Framework 4.8 installation can start.
>
> **Note:** Some running applications must be closed during the installation of .NET Framework 4.8, such as Server Manager.
>
> **Note:** After .NET Framework 4.8 is successfully installed, you must click **Restart** to restart the machine immediately. Otherwise, the upgrade will fail due to the failure of upgrading the new .Net Framework. After the machine reboots, ensure the FIPS is disabled after restarting the machine. To disable the FIPS setting, modify the **Enabled** value to **0** under the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy** directory of Windows registry.



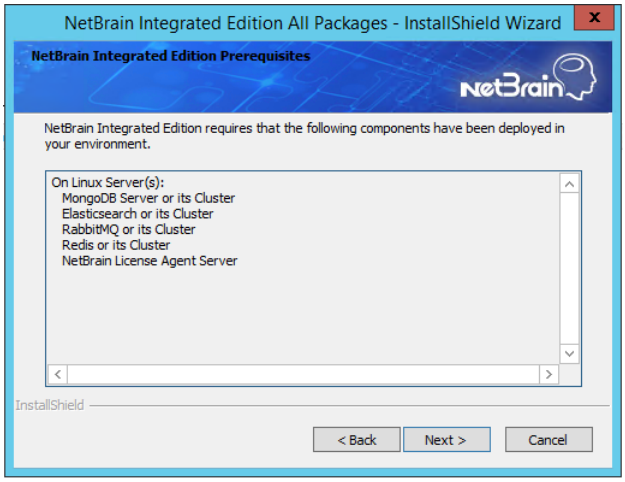
2) Stop all the NetBrain services on the Windows server manually before continuing the upgrade.

3) Click **Yes** in the dialog box to initiate the system upgrade.



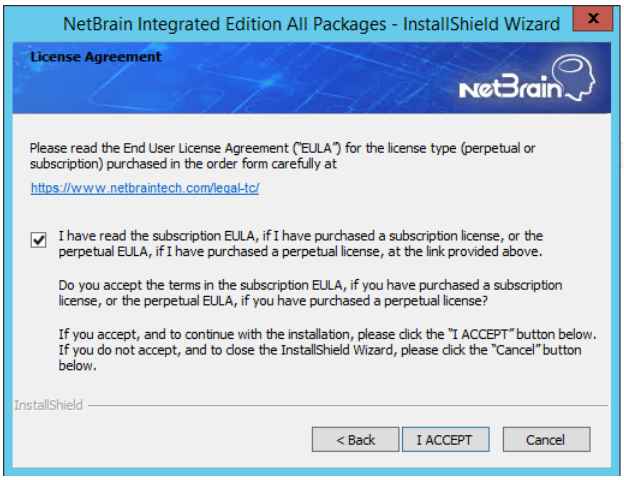
4) On the Welcome page, click **Next**.

5) On the NetBrain Integrated Edition Prerequisites page, read the list of Linux components that must be deployed beforehand in your environment and click **Next**.

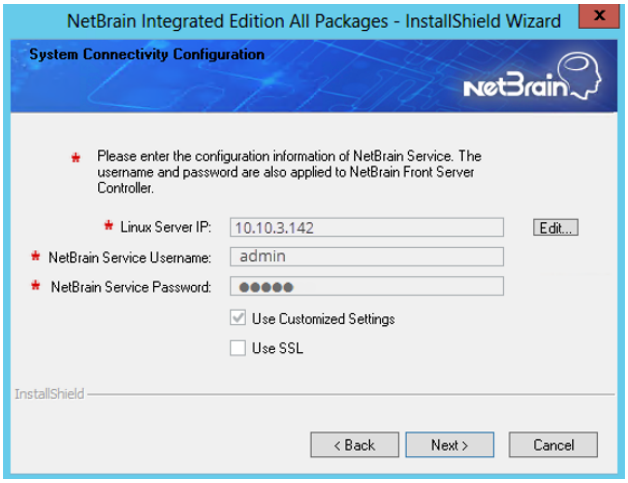


6) On the System Configuration page, review the system configuration summary and click **Next**.

7) On the License Agreement page, read the license agreement, and then select the **I have read the subscription EULA** check box and click **I ACCEPT**.

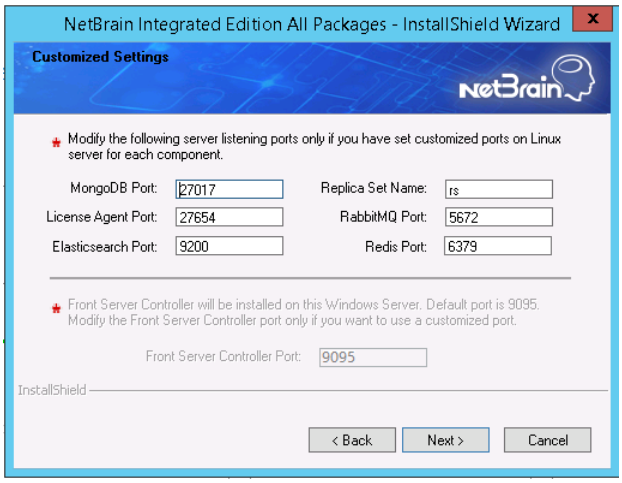


8) On the System Connectivity Configuration page, review the connection information to connect to NetBrain Database Server. Click **Next**.

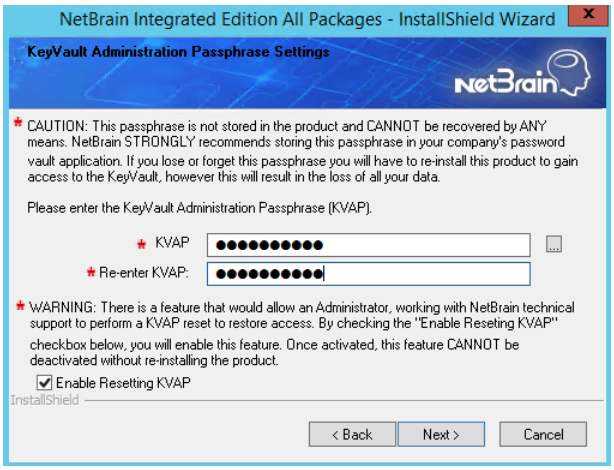


> **Note:** Checking the **Use SSL** option will inherit the SSL settings of your existing NetBrain Database Server.
>
> **Note:** If you customized a port number for any of Linux components during the installation of NetBrain Database Server, select the **Use Customized Settings** check box.

9) (Required only if the **Use Customized Settings** check box is selected) On the Customized Settings page, leave all fields unchanged. Click **Next**.

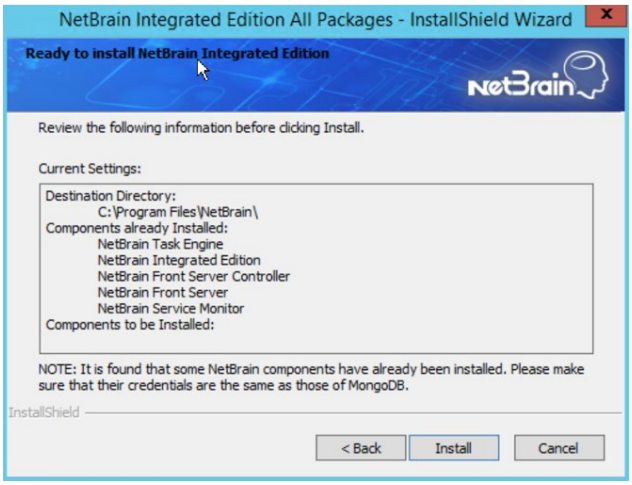


10) On the KeyVault Administration Passphrase Settings page, enter the existing passphrase twice and click **Next**.



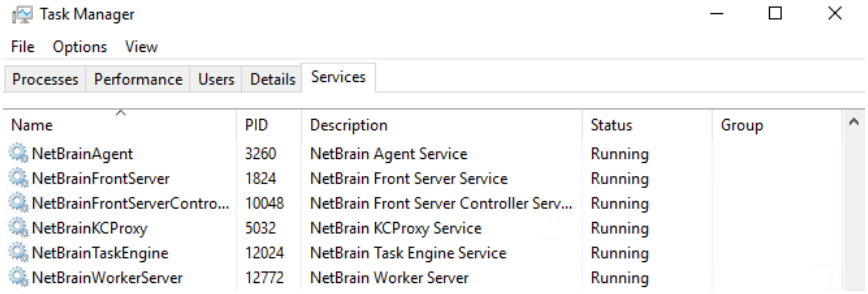
> **Note:** Keep notes of the passphrase because it is required when you scale up or upgrade the Application Server. In case of losing the passphrase, select the **Enable Resetting KVAP** check box so that NetBrain system admin can reset the passphrase at any time.

11) Review the server components to be upgraded and click **Install**. All the Windows components will be upgraded one by one.



5. After all components are successfully upgraded, click **Finish** to complete the upgrading process and exit the Installation Wizard.

6. Open the Task Manager and navigate to the **Services** tab, you can find the following running NetBrain services.

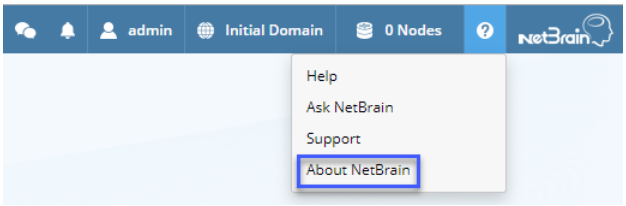


1.6. Verifying Upgrade Results

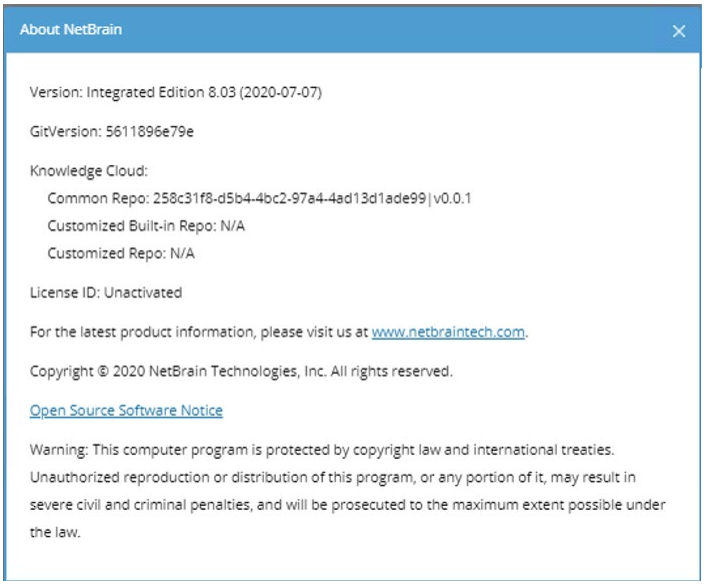
1. Do the following steps to check the IE version in web browser:

> **Note:** It is highly recommended to clear your web browser's cache before reloading the IE web page.

1) In the system Management page, click the icon and select **About NetBrain** from the quick access toolbar.



2) Check the version information.



2. Do the following steps to check the system version in MongoDB:

1) Log in to the Linux server where MongoDB is installed.

2) Open a command prompt and run the `mongo --host <IP or hostname of MongoDB Server:Port> -u <username> -p <password> --authenticationDatabase <database_name> --authenticationMechanism SCRAM-SHA-256` command to connect to MongoDB.

Example:

```
[root@localhost ~]# mongo --host 10.10.3.142:27017 -u mongodb -p mongodb --
authenticationDatabase admin --authenticationMechanism SCRAM-SHA-256
MongoDB shell version v4.0.6
connecting to: mongodb://10.10.3.142:27017/?authMechanism=SCRAM-SHA-
256&authSource=admin&gssapiServiceName=mongodb
...
```

> **Tip:** If SSL is enabled, run the `mongo --host <IP or hostname of MongoDB Server:Port> -u <username> -p <password> --authenticationDatabase <database_name> --ssl --sslAllowInvalidCertificates --authenticationMechanism SCRAM-SHA-256` command.

3) Run the `use NGSystem` command to switch to the **NGSystem** database.

```
rsnetbrain:PRIMARY> use NGSystem
switched to db NGSystem
```

4) Run the `db.SystemInfo.find({_id: "SystemVersion"})` command to check the system version number.

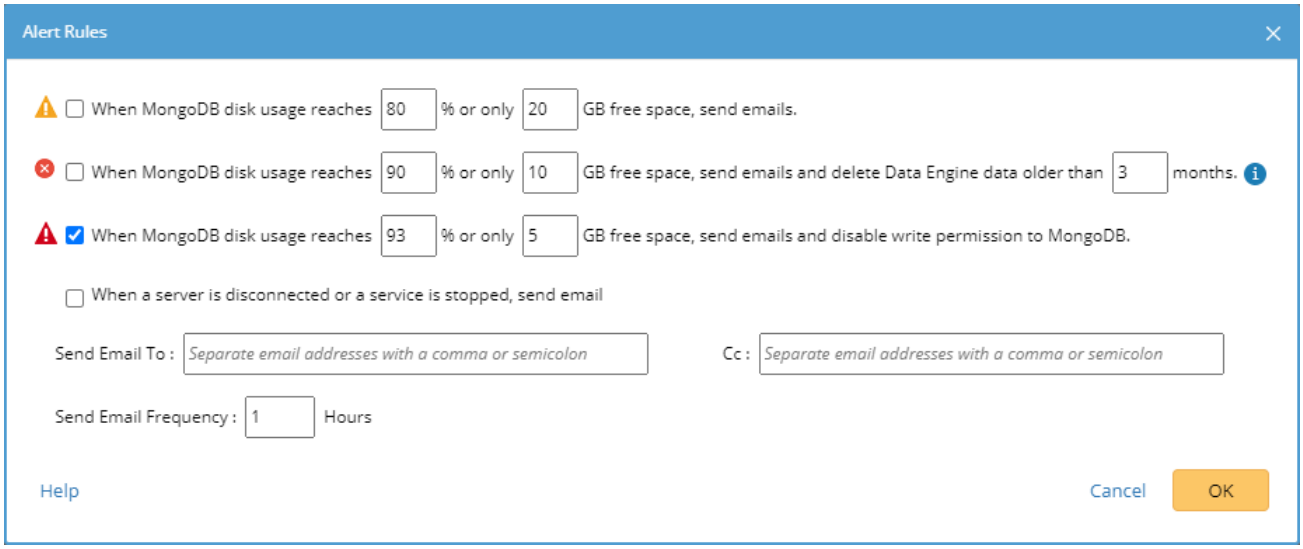
```
rsnetbrain:PRIMARY> db.SystemInfo.find({_id: "SystemVersion"})
{ "_id" : "SystemVersion", "version" : "8.0.03", "operateInfo" : { "opUser" : "NetBrain",
"opTime" :
  ISODate("2020-07-14T18:31:21.735") } }
```

5) Run the `exit` command to exit the command prompt.

1.7. Customizing MongoDB Disk Alert Rules

To proactively prevent the system database from data loss or even corruption, you can customize MongoDB disk alert rules with progressive quotas assigned. When the MongoDB usage reaches the predefined threshold, specified users can be notified by both email alerts and system alerts.

1. In the System Management page, click **Operations > Service Monitor** from the quick access toolbar.
2. In the Service Monitor home page, click **Alert Rules** at the upper-right corner. The default settings are as follows:



3. Change the settings based on your needs.
   1) Specify the disk usage threshold for different levels.

      > **Note:** To email alerts when a server is disconnected or a service is stopped, select the corresponding check box.

   2) Enter the email address in the **Send Email To** or **CC** fields.

      > **Note:** Email alerts are enabled only when email addresses are added at least in one field. Use a colon or semicolon to separate multiple items.

   3) Specify the frequency to send emails.
   4) Click **OK** to save the configuration.

1.8. Tuning Live Access

To tune live access, complete the following steps:

1. In your web browser, navigate to **http(s)://<IP address of NetBrain Application Server>/** to log in to your domain.
2. Click the domain name from the quick access toolbar and select **Manage Domain**.
3. In the Domain Management page, select **Operations > Advanced Tools > Tune Live Access** from the quick access toolbar. The **Tune Live Access** tab opens with all devices in the domain listed.
4. Click **Start Tuning**.
5. When the tuning process is completed, a notification message is displayed. Click **OK**.

1.9. Scheduling Benchmark Task

1. In the Domain Management page, select **Operations > Schedule Task** from the quick access toolbar.
2. On the **Schedule Task > Schedule Discovery/Benchmark** tab, select the **Enable** check box for the **Basic System Benchmark** entry.
3. Click the icon to select the **Run Now** option from the drop-down list to run the benchmark task immediately.

> **Note:** If you have multiple Front Servers, go to **Operations > Benchmark Tools > CheckPoint OPSEC Manager** to specify the target Front Server to access your CheckPoint firewalls and retrieve live data.

2. Appendix: Offline Installing Third-party Dependencies

1. Download the dependency package from a server with the Internet access using one of the following download links according to the version of your Operating System:
   - **CentOS7.5:** http://download.netbraintech.com/dependencies-centos7.5-8.0.tar.gz
   - **CentOS7.6:** http://download.netbraintech.com/dependencies-centos7.6-8.0.tar.gz
   - **CentOS7.7:** http://download.netbraintech.com/dependencies-centos7.7-8.0.tar.gz
   - **CentOS7.8:** http://download.netbraintech.com/dependencies-centos7.8-8.0.tar.gz
   - **CentOS7.9:** http://download.netbraintech.com/dependencies-centos7.9-8.0.tar.gz
   - **RHEL7.5:** http://download.netbraintech.com/dependencies-rhel7.5-8.0.tar.gz
   - **RHEL7.6:** http://download.netbraintech.com/dependencies-rhel7.6-8.0.tar.gz
   - **RHEL7.7:** http://download.netbraintech.com/dependencies-rhel7.7-8.0.tar.gz
   - **RHEL7.8:** http://download.netbraintech.com/dependencies-rhel7.8-8.0.tar.gz
   - **RHEL7.9:** http://download.netbraintech.com/dependencies-rhel7.9-8.0.tar.gz
2. Copy the downloaded dependency package to your Linux server.
3. Run the `tar -zxvf dependencies-<OS version>-8.0.tar.gz` command to decompress the package.

   > **Tip:** Possible values of **OS version** include: `centos7.5`; `centos7.6`; `centos7.7`; `centos7.8`; `rhel7.5`; `rhel7.6`; `rhel7.7`; `rhel7.8`.

4. Run the `cd dependencies` command to navigate to the decompressed directory.
5. Run the `offline-install.sh` command to install the dependencies.

3. Appendix: Editing a File with VI Editor

The following steps illustrate how to edit a configuration file with the vi editor, which is the default text file editing tool of a Linux operating system.

1. Create a terminal and run the `cd` command at the command line to navigate to the directory where the configuration file is located.
2. Run the `vi <configuration file name>` command under the directory to show the configuration file.
3. Press the **Insert** or **I** key on your keyboard, and then move the cursor to the location where you want to edit.
4. Modify the file based on your needs, and then press the **Esc** key to exit the input mode.
5. Enter the `:wq!` command and press the **Enter** key to save the changes and exit the vi editor.

4. Appendix: Restoring MongoDB Data

If you encounter data loss or corruption during the upgrade process, complete the following steps to restore MongoDB data with the backup data.

1. Log in to the Linux server where the MongoDB is installed as the **root** user.

2. Stop the MongoDB Service.
   1) Run the `systemctl stop mongod` command to stop the MongoDB service.
   2) Run the `ps -ef|grep mongod` command to verify whether the **mongod** process is stopped.

```
[root@localhost ~]# ps -ef| grep mongod
root     15136 14237  0 10:42 pts/2    00:00:00 grep --color=auto mongod
```

> **Note:** If the **mongod** process is stopped, the result should only contain one entry as shown above.

3. Restore the old data onto the MongoDB.
   1) Run the `cd /usr/lib/mongodb` command to navigate to the **/usr/lib/mongodb** directory.

> **Note:** If you modified the following default directory to store all MongoDB data files during the MongoDB installation, you must use the new directory (available in the **mongod.conf** file) accordingly.

   2) Run the `ls -al` command to browse all directories and files under the **/usr/lib/mongodb** directory.

```
[root@localhost mongodb]# ls -al
total 142
drwxr-xr-x. 5 netbrain netbrain   146 Oct 19 15:02 .
drwxr-xr-x. 4 root     root        42 Sep 19 14:41 ..
drwxr-xr-x. 4 root     root        42 Oct 19 15:03 data
drwxr-xr-x. 4 root     root       100 Oct 19 15: 03 log
-rwxr-xr-x. 2 netbrain netbrain  1004 Aug 25 17: 26 mongodb-keyfile
-rwxr-xr-x. 1 netbrain netbrain  1076 Oct 19 15:02 mongod.conf
```

   3) Run the `rm -rf ./data` command to delete the **data** directory.

   4) Run the `mv /etc/mongodb_databk/data` command under the **/usr/lib/mongodb** directory to move the data directory to the **/opt/mongodb** directory.

   5) Run the `ls -al` command to browse all directories and files under the **/usr/lib/mongodb** directory.

```
[root@localhost mongodb]# ls -al
total 142
drwxr-xr-x. 5 netbrain netbrain   146 Oct 19 15:02 .
drwxr-xr-x. 4 root     root        42 Sep 19 14:41 ..
drwxr-xr-x. 4 root     root     86016 Oct 19 15: 03 data
drwxr-xr-x. 4 root     root       100 Oct 19 15: 03 log
-rwxr-xr-x. 2 netbrain netbrain  1004 Aug 25 17: 26 mongodb-keyfile
```

```
-rwxr-xr-x. 1 netbrain netbrain  1076 Oct 19 15:02 mongod.conf
-rwxr-xr-x. 1 netbrain netbrain  1147 Oct 19 14:51 mongod.conf2017|Oct|19|10:15:50
```

4. Run the `systemctl mongod start` command to restart the MongoDB service.

5. Run the `mongo --host <IP or hostname of MongoDB Server:Port> -u <username> -p <password> --authenticationDatabase <database_name> --authenticationMechanism SCRAM-SHA-256` command to connect to the node.

   Example:

```
[root@localhost upgrade_replica_set]# mongo --host 10.10.3.142:27017 -u mongodb -p mongodb --
authenticationDatabase admin --authenticationMechanism SCRAM-SHA-256
MongoDB shell version v4.0.6
connecting to: mongodb://10.10.3.142:27017/?authMechanism=SCRAM-SHA-
256&authSource=admin&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0315bda2-73f3-4304-9166-c008b9b06ce3") }
MongoDB server version: 4.0.6
...
rsnetbrain:PRIMARY>
```

> **Tip:** If SSL is enabled, run the command `mongo --host <IP or hostname of MongoDB Server:Port> -u <username> -p <password> --authenticationDatabase <database_name> --ssl --sslAllowInvalidCertificates --authenticationMechanism SCRAM-SHA-256` command.

5. Appendix: Dumping MongoDB Data

The built-in MongoDB command `mongodump` is a simple and efficient tool for backing up a small volume of MongoDB data. However, for a large volume of data, it is more time-consuming than using the `cp` command introduced in Backing Up MongoDB Data.

> **Note:** Make sure the service of MongoDB is running when you run the `mongodump` command.
>
> **Note:** The dumped data can be used to restore data in any server.

1. Log in to the Linux server where the MongoDB is installed as **root** user.
2. Open a command prompt and run the `[root@localhost ~]# mkdir /etc/mongodb_databk` command to create a directory under the **/etc** directory to save the backup data.
3. Run the `mongodump --host <ip>:<port> -u <username> -p <password> --authenticationDatabase <database_name> --authenticationMechanism SCRAM-SHA-256 --gzip --out <filepath>` command to dump the MongoDB data to the **/etc/mongodb_databk** directory.

   Example:

   ```
   [root@localhost ~]# mongodump --host 127.0.0.1:27017 -u mongodb -p mongodb --
   authenticationDatabase admin --gzip --out /etc/mongodb_databk
   ```

   > **Tip:** If SSL is enabled, run the `mongodump --host <ip>:<port> -u <username> -p <password> --authenticationDatabase <dbname> --ssl --sslAllowInvalidCertificates --authenticationMechanism SCRAM-SHA-256 --gzip --out <filepath>` command.

4. Verify the backup result.
   1) Run the `cd /etc/mongodb_databk` command to navigate to the **/etc/mongodb_databk** directory.
   2) Run the `ls -al` command under the **mongodb_databk** directory to browse the backup data.

Restoring Dumped MongoDB Data

Restore the dumped data by using the `mongorestore` command provided by MongoDB.

> **Note:** Make sure the service of MongoDB is running when you run the `mongorestore` command.
>
> **Note:** Make sure other relevant services are stopped.

To restore the dumped MongoDB data onto the MongoDB server, run the `mongorestore --host <ip>:<port> -u <username> -p <password> --authenticationDatabase <database_name> --authenticationMechanism SCRAM-SHA-256 --gzip --out <filepath>` command.

Example:

```
[root@localhost ~]# mongorestore --host 127.0.0.1:27017 -u mongodb -p mongodb --
authenticationDatabase admin --gzip --out /etc/mongodb_databk
```

> **Tip:** If SSL is enabled, run the `mongorestore --host <ip>:<port> -u <username> -p <password> --authenticationDatabase <dbname> --ssl --sslAllowInvalidCertificates --authenticationMechanism SCRAM-SHA-256 --gzip --out <filepath>` command.