Backing Up and Restoring-Migrating NLS 2024

Trying to Migrate? See Migrating Nagios Log Server to a different server for steps on migrating without using the backup and restore method.

Before migrating, see the OS compatibility matrix.

Backup Overview

Nagios Log Server has several backup methods:

- Snapshots
  - Snapshots are backups of your Elasticsearch Log Data
- Config Snapshots
  - These are backups of the Inputs, Filters and Outputs for Logstash
- System Backups
  - This is a backup of the entire system (excluding the Elasticsearch log data)

Backups in Nagios Log Server (NLS) are slightly different than regular backup methods. This has to do with NLS being a cluster-oriented application. The backup methodology in NLS ensures that backups are held on all instances in the NLS cluster. This means that if an instance is lost you would not lose any data because the backup data is held on other NLS instances.

However, If you have a single instance cluster, then you will need to take additional manual steps to ensure that the config snapshots and system backups are stored on an external server. This will ensure you will be able to restore your single instance Nagios Log Server in the event of a disaster.

Snapshots

Snapshots are point-in-time backups of your log data that exists in the Elasticsearch database.

- Snapshots are stored in a Snapshot Repository
- The repository needs to be accessible by all instances in your Nagios Log Server cluster
    - Usually in a NFS or CIFS network share mounted to a path like /snapshot_repository
    - The mounted path needs to be identical on all instances

The snapshot is performed on the entire cluster. During the snapshot and maintenance job, a node will run the commands to create a new snapshot. Because the snapshot is of indexes that have shards allocated to different instances, you need an NFS or CIFS share so that those instances can store their data in the snapshot being created.

For more information on Snapshots, please refer to the following documentation:

Managing Snapshots And Maintenance

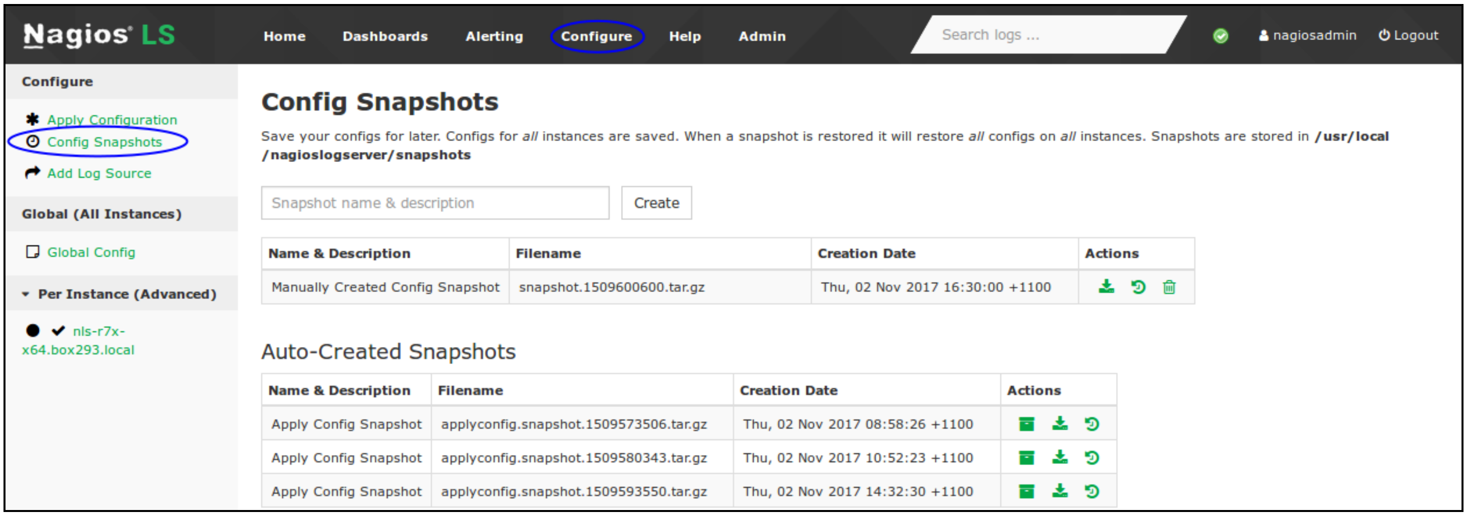
Again, If you have a single instance Nagios Log Server cluster, it is important that your snapshot repository is on another server. In this case, if your instance was to be completely destroyed you would would lose everything if it all was on the same server.

Config Snapshots

- Config Snapshots are backups of the Inputs, Filters and Outputs for Logstash. These are created automatically whenever you apply configuration or you can create manual config snapshots.

- Config snapshots are stored in the /usr/local/nagioslogserver/snapshots/ location on every instance in the cluster. Every instance in the cluster will have a copy of the config snapshots in this location ensuring that if an instance goes down the others have a copy of it.
- Config snapshots allow you to roll back to a point in time in the scenarios where you did not like the changes previously made. Manual snapshots will remain until you choose to delete them.

Config snapshots can be found by navigating to **Configure > Configure > Config Snapshots**.



Config Snapshots

Save your configs for later. Configs for *all* instances are saved. When a snapshot is restored it will restore *all* configs on *all* instances. Snapshots are stored in **/usr/local/nagioslogserver/snapshots**

Name & Description	Filename	Creation Date	Actions
Manually Created Config Snapshot	snapshot.1509600600.tar.gz	Thu, 02 Nov 2017 16:30:00 +1100	⬇ ↺ 🗑

Auto-Created Snapshots

Name & Description	Filename	Creation Date	Actions
Apply Config Snapshot	applyconfig.snapshot.1509573506.tar.gz	Thu, 02 Nov 2017 08:58:26 +1100	🗃 ⬇ ↺
Apply Config Snapshot	applyconfig.snapshot.1509580343.tar.gz	Thu, 02 Nov 2017 10:52:23 +1100	🗃 ⬇ ↺
Apply Config Snapshot	applyconfig.snapshot.1509593550.tar.gz	Thu, 02 Nov 2017 14:32:30 +1100	🗃 ⬇ ↺

Here you can see the existing snapshots that exist.

Manual and auto-created snapshots have these actions available:

- ⬇This allows you to download the .tar.gz file to your computer
- ↺This allows you to restore the snapshot to all instances in the cluster

Auto-created snapshots have this action available:

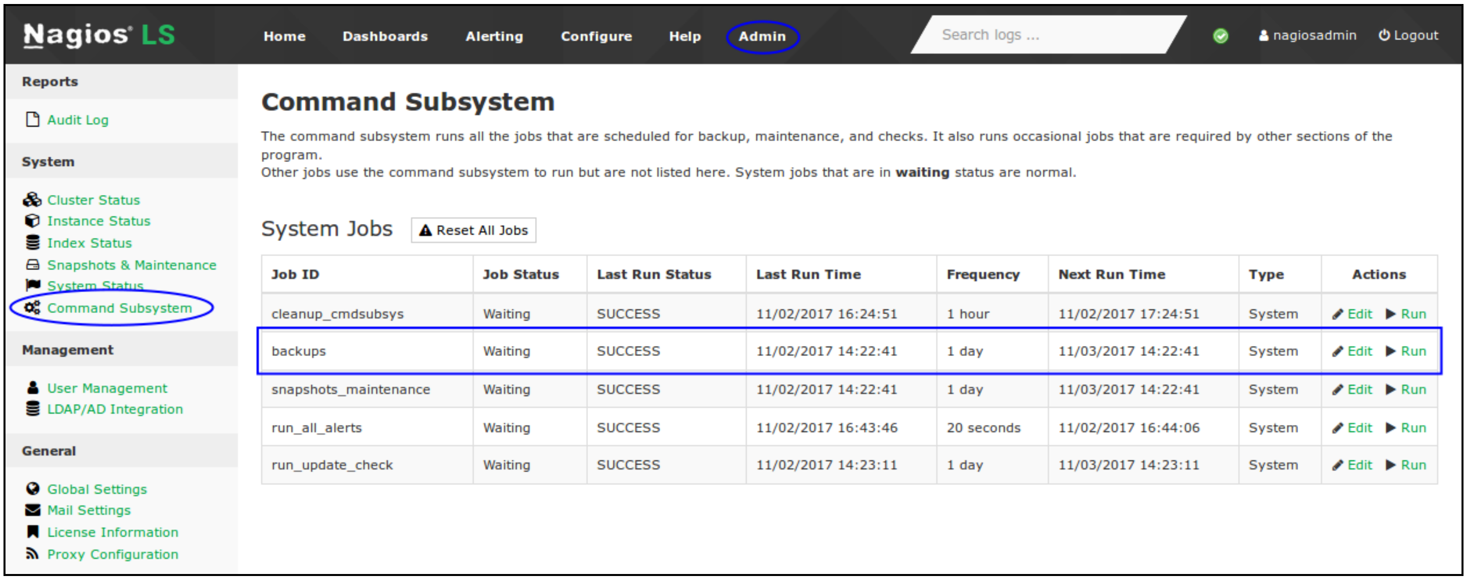
- 🗃This allows you to archive the Auto-Created snapshot to the manual snapshots section above

Manually Created / Archived snapshots have this action available:

- 🗑This allows you to permanently delete the snapshot

System Backups

- System backups contain the configuration settings, dashboards, users, internal logs, alerts. Also included are the Inputs, Filters and Outputs for Logstash.
- They are stored in the location /store/backups/nagioslogserver/ and are a file named based on the current date and epoch value, for example nagioslogserver.2017-05-09.1494303122.tar.gz.
- These backups are stored on every instance in your log server cluster. Whenever the backup job is scheduled to run, each instance will create a local copy of the backup. This means that if you were to lose a instance in your cluster, another instance will have a copy of this backup. This however does not protect you if all of your instances were to be lost in a disaster. You should periodically take a copy of the system backup to an external location to ensure you can restore Nagios Log Server.
- System backups are configured to run once a day as a system job. By default the time they are run is based on when you installed the first instance in your Nagios Log Server cluster. Navigate to Admin > System > Command Subsystem and you will find the backups system job. From here you can change the frequency of the snapshot and also initiate one to run now.



Command Subsystem

The command subsystem runs all the jobs that are scheduled for backup, maintenance, and checks. It also runs occasional jobs that are required by other sections of the program.
Other jobs use the command subsystem to run but are not listed here. System jobs that are in **waiting** status are normal.

System Jobs

Job ID	Job Status	Last Run Status	Last Run Time	Frequency	Next Run Time	Type	Actions
cleanup_cmdsubsys	Waiting	SUCCESS	11/02/2017 16:24:51	1 hour	11/02/2017 17:24:51	System	Edit ▶ Run
backups	Waiting	SUCCESS	11/02/2017 14:22:41	1 day	11/03/2017 14:22:41	System	Edit ▶ Run
snapshots_maintenance	Waiting	SUCCESS	11/02/2017 14:22:41	1 day	11/03/2017 14:22:41	System	Edit ▶ Run
run_all_alerts	Waiting	SUCCESS	11/02/2017 16:43:46	20 seconds	11/02/2017 16:44:06	System	Edit ▶ Run
run_update_check	Waiting	SUCCESS	11/02/2017 14:23:11	1 day	11/03/2017 14:23:11	System	Edit ▶ Run

- There is no location in the Nagios Log Server GUI to view the system backups, you will need to establish a terminal session to a Nagios Log Server instance and check the directory to ensure the backups exist in this location.

What Happens In A Disaster

Multiple Instance Cluster - Losing One Instance

When you have multiple instances in your Nagios Log Server cluster and you lose one of those instances, generally the impact is minimal.

- The cluster will continue to function as the Elasticsearch data is spread across the instances
- Any log data that is being sent to the down instance will not be received
    - If the log data is sent to a load balancer then it will be diverted to another instance (not a part of Nagios Log Server)

To return the cluster back to a healthy state you can attempt to repair the problem that caused the instance to fail in the first place (in the case of a physical hardware failure). Once the instance re-connects to the cluster Elasticsearch will automatically become updated with the rest of the data in the cluster.

If you have devices sending log data to this instance:

- If you expect the instance to be down for an extended period of time you should reconfigure the devices to send their data to another instance
- If updating every device to send data to another instance is too time consuming then you can instead run up a fresh install of Nagios Log Server that uses the existing IP address of the old instance and add it to the existing cluster

It's worth mentioning that any one instance in the cluster is no more important than another. If you lose a instance due to problems that require a substantial amount of time to repair, it can be simpler and quicker to run up a fresh install of Nagios Log Server that uses the existing IP address of the old instance and add it to the existing cluster.

Multiple Instance Cluster - Losing Multiple Instances

The impact is very similar to just losing one instance. This has to do with how the Elasticsearch data is spread across the instances. Follow the same principles outlined in the previous section.

Single Instance Cluster OR Multiple Instance Cluster - Losing ALL Instances

This scenario is more likely going to occur when you have a single instance cluster. It is possible to lose all instances in a multi-instance cluster and in that scenario the recover steps are the same for the single instance cluster.

- Run up a fresh install of Nagios Log Server on a new instance
- Restore the System Backup

- Mount the Snapshot Repository
- Restore the indexes to recover the log data

In a multi-instance scenario, once you have the first instance running then it is a simple matter of running up additional instances and adding them to the cluster.

Single Instance OR Multiple Instance Cluster - Losing ALL Instances and NO SYSTEM BACKUP

This is the worst case scenario, it is more likely going to occur when you have a single instance cluster. It is possible to lose all instances in a multi-instance cluster and in that scenario the recover steps are the same for the single instance cluster.

Not having a copy of the System Backup requires additional manual steps to recover. All of the data in the System Backup is also included in the snapshots, so as long as you have your snapshot repository available then you will still be able to recover.

- Run up a fresh install of Nagios Log Server on a new instance and install it as a new cluster
- Mount the Snapshot Repository
- Restore the recent snapshots of kibana-int, nagioslogserver and nagioslogserver_log
- Restore the indexes to recover the log data
- Reset the backend jobs

In a multi-instance scenario, once you have the first instance running then it is a simple matter of running up additional instances and adding them to the cluster.

Restoring A System Backup or Migrating to a different Server

If you need to follow these steps it is assumed that you have lost all instances in your cluster, (single instance or multiple instance). You should not follow these steps if you have lost an instance in a multi instance cluster and need to repair it, these steps do not apply.

You can also follow these steps if you want to test restoring a system backup. You do not need to perform this action in an isolated network as the restored cluster has a different ID and won't affect your production cluster.

Fresh Install Of Nagios Log Server

The first step is to run up a fresh install of Nagios Log Server. This can be on the existing hardware of a instance that died however it is recommended that you perform a clean install of the RHEL or CentOS operating system.

Perform the steps in the following documentation:

Nagios Log Server - Manual Installation Instructions

Once the install is complete DO NOT navigate to the user interface to complete the final installation steps. Leave the terminal open as you will use it in the following steps.

Restore The System Backup

Next you will need to transfer your system backup to the /store/backups/nagioslogserver/ directory on this instance.

You can use a program like WinSCP to do the transfer or use another method like scp.

To restore the system backup execute the following commands in the terminal session:

```
cd /usr/local/nagioslogserver/scripts/
```

```
./restore_backup.sh /store/backups/nagioslogserver/nagioslogserver.2017-05-10.1494373596.tar.gz
```

You can see that the backup file used in this example is nagioslogserver.2017-05-10.1494373596.tar.gz, you will need to change this to match the name of your system backup.

You will see the message "Restore Complete!" when it has finished. At this point you should open the web GUI to this instance and login to check that it is OK. Dashboards, inputs, filters, users and other settings should exist however there will not be any log data available to query against, this will be covered next.

Mount The Snapshot Repository

Now mount the snapshot repository that contains your existing snapshots. In this example it will be mounted to /snapshot_repository.

Once the repository is mounted, open the web GUI and navigate to Admin > System > Snapshots & Maintenance.

- Click the Create Repository button and populate the fields for the new repository
- The Location field will be /snapshot_repository in this example
- Click the Add Repository button to create the repository

Now that the repository has been created, the Snapshots list will be populated with the existing snapshots that can be restored.

Restore Indexes

To restore the existing log data you need to restore the indexes from your snapshot repository.

- Click the Restore icon to restore the required snapshot
- Select the indexes that you want to restore and then click the Restore Indexes button
- The restore process will run in the background
- To confirm they have been restored navigate to Admin > System > Index Status

This completes the process of restoring Nagios Log Server from a system backup. At this point you can add more instances to the cluster if required.

Restoring or migrating WITHOUT A System Backup

If you need to follow these steps it is assumed that you have lost all instances in your cluster, (single instance or multiple instance) AND you do not have a copy of your system backup. You should only follow these steps in a worse case scenario.

You can also follow these steps if you want to test this method. You do not need to perform this action in an isolated network as the restored cluster has a different ID and won't affect your production cluster.

Fresh Install Of Nagios Log Server

The first step is to run up a fresh install of Nagios Log Server. This can be on the existing hardware of a instance that died however it is recommended that you perform a clean install of the RHEL or CentOS operating system.

Perform the steps in the following documentation:

Nagios Log Server - Manual Installation Instructions

Once the install is complete navigate to the user interface and complete the final installation steps. You need to have a functioning Nagios Log Server cluster for this method to work. Leave the terminal open as you will use it in the following steps.

Mount The Snapshot Repository

Now mount the snapshot repository that contains your existing snapshots. In this example it will be mounted to /snapshot_repository.

Once the repository is mounted, open the web GUI and navigate to Admin > System > Snapshots & Maintenance.

- Click the Create Repository button and populate the fields for the new repository
- The Location field will be /snapshot_repository in this example
- Click the Add Repository button to create the repository

Now that the repository has been created, the Snapshots list will be populated with the existing snapshots that can be restored. DO NOT attempt to restore any indexes as this point.

```
Restore kibana-int, nagioslogserver and nagioslogserver_log
```

These steps will restore everything that would have been restored normally using a system backup. These require steps to be executed in your terminal session.

First you need the name of the snapshot repository by executing this command:

```
curl -XGET "localhost:9200/_snapshot?pretty"
```

This should output something like:

```
{
 "snapshot_repository" : {
  "type" : "fs",
  "settings" : {
   "compress" : "true",
   "location" : "/snapshot_repository"
  }
 }
}
```

The name snapshot_repository is what is required for the next command. This next command will show all available snapshots taken in the past 5 days:

```
curator show snapshots --repository snapshot_repository --newer-than 5 --
time-unit days
```

Here is an example of the output:

```
2017-05-10 16:47:21,198 INFO      Job starting: show snapshots
2017-05-10 16:47:21,219 INFO      Matching snapshots:
curator-20170509213607
curator-20170508714688
curator-20170507212685
curator-20170506318623
curator-20170505263402
```

The snapshot that is going to be targeted in this example is curator-20170509213607. Now we need to confirm that this snapshot contains the required indexes. The following command requires the name of the repository snapshot_repository and the snapshot curator-20170509213607:

```
curl -XGET 'localhost:9200/_snapshot/snapshot_repository/curator-
20170509213607?pretty'
```

The output will be something like:

```
{
  "snapshots" : [ {
    "snapshot" : "curator-20170509213607",
    "version_id" : 1070699,
    "version" : "1.7.6",
    "indices" : [ "kibana-int", "logstash-2017.05.09", "nagioslogserver", "nagioslogserver_log" ],
    "state" : "SUCCESS",
    "start_time" : "2017-05-09T21:36:07.261Z",
    "start_time_in_millis" : 1494365767261,
    "end_time" : "2017-05-09T21:36:07.976Z",
    "end_time_in_millis" : 1494365767976,
    "duration_in_millis" : 715,
    "failures" : [ ],
    "shards" : {
      "total" : 16,
      "failed" : 0,
      "successful" : 16
    }
  } ]
}
```

Specifically this line is what you are after:

"indices" : ["kibana-int", "logstash-2017.05.09", "nagioslogserver", "nagioslogserver_log"],

In tells us that it has recent snapshots of the kibana-int, nagioslogserver and nagioslogserver_log indexes.

Now that you have confirmed the repository contains the indexes required you can now restore them. Each index requires two commands to perform a restore. The first command closes the index. The second command restores the index (and automatically re-opens it). These commands are similar to the previous ones where they require the name of the repository and snapshot to restore from.

This command closes the kibana-int index:

```
curl -XPOST 'localhost:9200/kibana-int/_close?pretty'
```

Which should output:

```
{
  "acknowledged" : true
}
```

This command restores the kibana-int index (it's one long command that wraps over two lines):

```
curl -XPOST 'localhost:9200/_snapshot/snapshot_repository/curator-
20170509213607/_restore?pretty' -d '{ "indices":"kibana-int"}'
```

Which should output:

```
{
  "accepted" : true
}
```

The following commands will restore the nagioslogserver and nagioslogserver_log indexes:

```
curl -XPOST 'localhost:9200/nagioslogserver/_close?pretty'
curl -XPOST 'localhost:9200/_snapshot/snapshot_repository/curator-
20170509213607/_restore?pretty' -d '{ "indices":"nagioslogserver"}'
```

```
curl -XPOST 'localhost:9200/nagioslogserver_log/_close?pretty'
```

```
curl -XPOST 'localhost:9200/_snapshot/snapshot_repository/curator-
20170509213607/_restore?pretty' -d '{ "indices":"nagioslogserver_log"}'
```

Once these commands have been executed you should now refresh the Nagios Log Server interface. Navigate around to confirm that your dashboards, inputs, filters, users and other settings have been restored. There will not be any log data available to query against, this will be covered next.

Restore Indexes

To restore the existing log data you need to restore the indexes from your snapshot repository.

- Navigate to Admin > System > Snapshots & Maintenance
- Click the Restore icon to restore the required snapshot
- Select the indexes that you want to restore and then click the Restore Indexes button
- The restore process will run in the background
- To confirm they have been restored navigate to Admin > System > Index Status

This completes the process of restoring or migrating Nagios Log Server without a system backup. At this point you can add more instances to the cluster if required.