Nagios XI – Resizing the VM Disk Size

Nagios®

The Industry Standard in IT Infrastructure Monitoring

Purpose

This document describes how to increase the size of the Nagios XI virtual machine (VM) disk size.

Target Audience

This document is intended for use by Nagios XI Administrators who require more space on their XI installation and are using the virtual machine image.

Preparatory steps

The actions necessary to resize the virtual machine are potentially destructive by nature, so while things should work without problems it is strongly recommended that you make proper backups of your installation before proceeding.

Resizing the virtual disk

The first step is to modify the amount of disk space VMware (or other virtual machine software) is allocating to your XI image.

With VMware ESX Server this step can be done with the VM running. The setting you need can be found under
(Your VM) → **Summary** → **Commands** → **Edit Settings** → **Hardware** → **Hard Disk 1** → **Capacity** → **New Size**.

For VMware Player, if on the overview page select
(Your VM) → **Edit virtual machine settings** → **Hardware** → **Hard Disk (SCSI)** → **Utilities** → **Expand** → **Maximum disk size**,
and if the VM is running start with **VM** → **Settings**.

Resizing the partition

For our task at hand, the next step is to modify the partition table within the system to recognize that the disk size has changed. This can be done using the *fdisk* utility, which is a partition editor. For the stock virtual machine, *sda1* is used for the boot partition and *sda2* is the physical device for all logical volumes – we want to edit the latter. To do so, open the parent device, *sda*, for editing:

```
# fdisk /dev/sda
```

You probably want to start by printing the current partition table, so you can keep track of your changes as you go.

```
Command (m for help): p
```

To resize a partition, you actually “delete” it and then add a new one with the new size, making sure to place the beginning of the new partition at the same sector/cylinder as the old one started.

```
Command (m for help): d
Partition number (1-4): 2
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
```

You should be able to accept the defaults for the start and end point – just double-check that they make sense when compared to the information printed earlier. Then, make sure the partition type is set correctly:

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Linux LVM)
```

At this point you are done making your changes, and just need to write them to the disk. It may be a good idea to use p to print the new table just to be sure, then commit the table:

```
Command (m for help): w
The partition table has been altered!
```

You will now need to reboot the virtual machine to make the kernel recognize the new partition table.

Resizing the physical volume (PV)

The CentOS system installed in the virtual machine image we ship uses *Logical Volume Management (LVM)* for its partitions. If you have never worked with LVM before and want to better understand the terminology herein, take a look at *http://en.wikipedia.org/wiki/Logical_volume_management* and *http://www.tldp.org/HOWTO/LVM-HOWTO/*.

In order to make use of this space, LVM needs to be told that it is there and should be used. For this you will adjust the physical volume to match the partition. Since we just want the physical volume to automatically expand to use whatever free space is available on the device, the command is simple.

```
# pvresize /dev/sda2
```

Resizing the logical volume (LV)

Next is resizing the logical volume to use the new space available in the LVM physical volume. Again, since we are using all of it there is an automatic option.

```
# lvresize /dev/mapper/VolGroup00-LogVol00 /dev/sda2
```

Resizing the filesystem

Lastly, you will need to change the actual filesystem residing on the logical volume so that it knows to put files in that space. Once more, we will take advantage of automatically assumed options since we are using all available space.

```
# resize2fs /dev/mapper/VolGroup00-LogVol00
```

You can confirm your work with df -h, which will display a summary of filesystem size and usage.