Linux CentOS 10 Installation

NetVizura needs dedicated server

For security reasons, make sure your server or VM has no software installed before installing NetVizura. Other software of services running on the same server can impact installation.

NetVizura needs correct time

Before installing NetVizura, make sure the time on your server is set correctly. Time change after the installation will invalidate the license!

NetVizura installation needs internet access

NetVizura requires an active internet connection to install the required dependent software. After the installation is successful, you can disable internet access for the NetVizura server.

On this page:

Installation Steps

To install NetVizura, you will need root privileges.

To install NetVizura, follow these steps:

Step 1: Download and execute the CentOS 10 prerequisites script:

```
su -
dnf install wget -y
wget https://www.netvizura.com/files/products/general/downloads/netvizura-6.1-prerequisites-centos10.sh --output-document=/tmp/netvizura-prerequisites-centos10.sh
bash /tmp/netvizura-prerequisites-centos10.sh
```

Prerequisites script will install required dependent software:

- Wget
- Tomcat 9
- PostgreSQL 5

Step 2: Installing the NetVizura package
Download NetVizura RPM package from NetVizura website to NetVizura server's `/tmp` directory and execute the following command:

```
dnf -y localinstall /tmp/netvizura-package.rpm
```

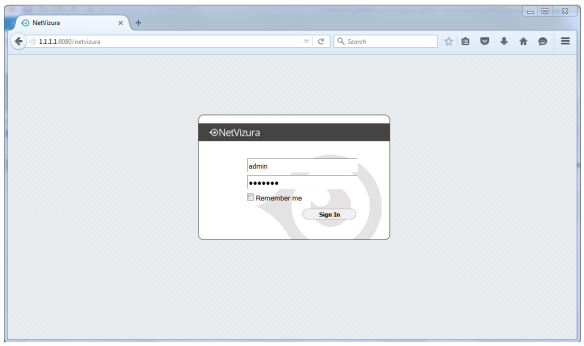
Step 3: Verify installation

Now you can go to NetVizura web interface *http://<netvizura_server_ip>:8080/netvizura*.

Default login credentials:

- Username: **admin**
- Password: **admin01**

For example, if your server IP is 1.1.1.1, then point your browser to *http://1.1.1.1:8080/netvizura,* like in the screenshot below:



Post Install Steps

After installation, configuration files must be tweaked to utilize the installed RAM to the fullest extent. The main consumers of RAM are the operating system, PostgreSQL/Elasticsearch database, and Tomcat. The general rule for allocating memory is to split it in a 2:1 ratio between PostgreSQL/Elasticsearch and Tomcat, with 1 GB or more reserved for the operating system.

For instance:

Installed RAM	PostgreSQL	Tomcat	OS	Elasticsearch
6 GB	2 GB	1 GB	1 GB	2 GB
16 GB	2 GB	5 GB	1 GB	5 GB

Tweaking PostgreSQL

Tweaking PostgreSQL for optimal performance is a topic on which many books have been written, but the following are some common-sense suggestions. For the curious ones, recommended reads (among countless others) include the PostgreSQL Optimization Guide and the PostgreSQL Tuning Guide.

In order to apply the following tweaks, edit the file `/var/lib/pgsql/PG_VERSION_NUMBER/data/postgresql.conf`. You will need to restart the PostgreSQL service after editing with the command: `service postgresql restart`. Almost all of the following parameters are commented with a carron character (#). Be aware that if you comment out the changed parameter, PostgreSQL will revert to the default value.

In the following example, it is assumed that 4 GB of RAM is allocated for PostgreSQL.

> ⚠ Before changing any parameters in PostgreSQL configuration, read the provided comments in the table below for more information regarding a specific parameter.

parameter	recommended value	comment
`max_connections`	30	NetVizura rarely uses more than 10 connections simultaneously, but it is good to have some reserve.
`shared_buffers`	1024MB	The recommended amount is `RAM/4`.
`effective_cache_size`	2048MB	The recommended amount is `RAM/2`, possibly even `RAM * 3/4`.
`checkpoint_completion_target`	0.7	This parameter can take values between 0 and 1. The default is 0.5, meaning the write phase of the checkpoint process will take half of the checkpoint timeout. Increasing this value will give the checkpoint write phase more time to complete, thereby reducing I/O usage.
`work_mem`	32-64MB	The formula used is `max_connections*work_mem <= RAM/4`, but using a bit more is still fine.
`maintenance_work_mem`	256MB	Speeds up DB self clean process. Usually 4*work_mem or something in that ballpark

`wal_buffers`	16MB	Increasing `wal_buffers` is helpful for write-heavy systems. Usually, this is 16MB.
`min_wal_size`	1GB	If WAL files are under this size, files will be recycled for future checkpoints.
`max_wal_size`	2GB	The maximum size of WAL files, after which the CHECKPOINT command is issued, and files are written to disk.
`effective_io_concurrency`	2	The number of simultaneous requests that can be handled efficiently by the disk subsystem.
`full_page_writes`	off	Turning this parameter off speeds up normal operation but might lead to unrecoverable or silent data corruption after a power outage, OS failure, or HDD failure. The risks are similar to turning off `fsync`, though smaller.
`fsync`	off	Don't wait for the HDD to finish the previous *write* operation. This brings the most benefit, but if there is a power outage, an OS or HDD failure at the exact instant when PSQL issues a write command to the HDD, that data will be lost, and the DB itself could be corrupted. On the other hand, DB can issue several times as many write commands in the same time period and consider them all done, thus improving write performance immensely.
`synchronous_commit`	off	Similar to "fsync" but with less benefit.
Parallel system optimization (PSQL => 9.6)		
`max_worker_processes`	2	Number of cores
`max_parallel_workers_per_gather`	1	Number of cores/2
`(PSQL > 9.6) max_parallel_workers`	2	Number of cores

Tomcat Memory Allocation

During installation, NetVizura automatically allocates memory for the Tomcat process. The amount allocated to the Tomcat process is calculated according to the formula:

$(RAM_{total}$ - 1GB) / 3 but no less than 1GB.

For instance:

Total RAM	Tomcat
3 GB	1 GB
4 GB	1 GB
16 GB	5 GB

However, if you need to tweak Tomcat RAM allocation differently (the example for 2048MB):

1. Edit file `/etc/tomcat/conf.d/netvizura.conf`
2. Locate `JAVA_OPTS` environment variable that defines memory. This line looks something like the following:
   `JAVA_OPTS="${JAVA_OPTS} -Xmx1024m -Xms1024m"`
3. Modify the `-Xmx` and `-Xms` to the same amount. This should look something like:
   `JAVA_OPTS="${JAVA_OPTS} -Xmx2048M -Xms2048M"`

Save the file and restart Tomcat: `systemctl restart tomcat.service`

Elasticsearch Memory Optimization

By default, the memory limit for Elasticsearch should be set to 30% of RAM. If you need it set to a different value, edit the file/etc/elasticsearch/jvm.options and set the -Xms and Xmx values to the desired size. Then, restart the Elasticsearch and Tomcat services.