NetYCE System Requirements

updated: October 2016

Overview

Linux Operating System

NetYCE 7 uses the Redhat Enterprise Linux x86, 64-bit architecture, or RHEL X86_64 for short. Specifically the latest version of 6.x.

Where we state RHEL, we also include CentOS which is the non-commercial version of Redhat RHEL.

Redhat released version 7 of RHEL some time ago and many data centers support both version 6.x and 7.x. Currently NetYCE is not supported on RHEL 7 which will be rectified with the next release (NetYCE 7.1), expected in 2017.

NetYCE support for IPv6 communication with both users (front-end) and devices (management) is dependent on RHEL version 7.x as is outlined in the table below:

Feature	**NetYCE 6.3 RHEL 6.8**	**NetYCE 7.0 RHEL 6.8**	**NetYCE 7.1 RHEL 6.8**	**NetYCE 7.1 RHEL 7.x**
IPv6 modeling	no	yes	yes	yes
IPv6 device communication	no	no	no	yes
IPv6 dual-stack	no	no	no	yes

The RHEL and NetYCE combination is fully supported as a virtual VMware server. Customers can create the RHEL/CentOS system based on their hardening and management policies, or can opt to use the VM NetYCE provides.

A NetYCE deployment can consist of a single server or a multi-tier multi-server architecture using front-end and database servers in a high-availability configuration. The operating system and software requirements for any NetYCE system regardless of its role, are identical. Actually, NetYCE server roles and their relationships can be altered in minutes using a configuration tool.

MySQL Database

NetYCE uses the database MariaDB version 10.0.x. MariaDB is based on the MySQL database that is currently owned by Oracle. MariaDB offers enhanced replication features that MySQL lacks and comes without the Oracle licensing.

MariaDB can be installed by NetYCE during the initial software installation or by the customer as desired.

NetYCE packages

The NetYCE software distribution comes in two packages (apart from some required standard available RHEL/CentOS packages). These packages are customer installable although we prefer to perform the initial installation in collaboration with a local administrator.

Both packages can be downloaded from the NetYCE Wiki-documentation site. The larger of the two, “YcePerl_<version>” is normally only updated on major releases. The other package, “Yce_<version>_<patchlevel>”, forms the actual software distribution which can used to perform a software update using the NetYCE web-GUI. A process requiring a few minutes per server.

Software distribution updates are released at least once a month and include ongoing product features and fixes.

See for more details: Download Patchfiles

local Wiki

Optionally, a local copy of the NetYCE Wiki documentation pages can be installed on one or more of the NetYCE servers.

This option is useful when the NetYCE servers cannot access the Internet - which in all honesty should never be possible for any NMS system - or the user cannot access the public Wiki server due to Internet restrains or lacking the required login account.

If the required 'mod_php' module for the http server (Apache) is installed, two additional packages can be downloaded from the Wiki: The Wiki engine and the Wiki distribution.

The Wiki engine needs to be installed only once and can be performed through the Web-GUI. The Wiki distribution is updated daily on the public wiki server which is then also updated through the Web-GUI.

After initial installation of the Wiki engine, some additional configuration changes need to be made to the system. It also requires a DNS alias pointing to the NetYCE server to be available.

See for more details: Download WIKI installation files

RHEL packages

Although a standard RHEL server installation requires many, many different packages, some are a 'must' for NetYCE. They may or may not be installed depending on selected options during installation.

The following packages are required by NetYCE:

– this list is not complete! (yet) –

```
httpd (apache 2)
```

```
mod_php (optional - for local NetYCE Wiki support)
mod_ssl (optional - but highly recommended since https and tls support
depends on it)
openssh
telnet client
fping (fastping ipv4, non RHEL available from NetYCE)
fping6 (fastping ipv6, NetYCE v7.1 and later)
vsftpd (optional - to allow devices to exchange files with the NetYCE
server)
tftp client (optional - to connect to local tftp server for testing
purposes)
```

SElinux support

We strongly advise NOT to activate the “SElinux” security manager. From past experience we learned that many unexplained failures were caused by incorrect permissions enforced by SElinux. Since these failures lack (accessible) error messages, inordinate amounts of time are spent in attempting to reproduce and fix these non-existent errors.

All NetYCE software IS compatible with an enforcing SElinux, but usually the required system administrators skills fall short to implement the policies properly and consistently.

System Specifications

The hardware requirements of NetYCE are moderate by itself although much depends on the intended level of use and the application architecture selected.

The NetYCE architecture uses three basic setups: single-server, high-availability, and multi-tier. These options are illustrated in some detail in the article YCE Connection matrix

In general we suggest to deploy two NetYCE servers in different data centers attached to Network Management (NMS) networks. This high-availability setup will provide both front-end (user and network facing) functions AND database functions per server. These functions can be configured to provide live failover and backup services by means of master-master replication. The front-end functions support 10-20 simultaneous users and can execute several thousand config changes per hour.

For such deployments a physical or virtual x86 server needs to have at least two CPU cores and 4 GB of memory, but 4 cores and 8 GB memory is recommended.

Memory usage is primarily determined by database size and the number of desired parallel jobs. Systems running both high job-loads and a large database should consider installing 16 GB memory. Additional cores will improve mostly the systems perceived 'responsiveness'.

When the multi-tier architecture is selected, each of the server specification can be tuned to their role: database only, combined front-end (user and networking), user front-end only, and networking only. For higher concurrent users and higher job-loads, memory is the prime requirement. Likewise for a larger database, more memory is needed as does the number of concurrent users. However, a large database is caused more by high port and subnet count than by high node count.

Disk space can be local or SAN based and generally will not exceed 50 GB. This disk space is allotted to a single filesystem or split across several, depending on system management preferences.

The NetYCE directory structure uses several trees for various functions. Assigning the mysql, shared and working+logs trees to individual filesystems is recommended.

```
/ - 3 to 6 GB (OS root, bin, usr, lib, opt, etc)
/opt/yce - 100 MB (netyce binaries)
/opt/nms - 100 MB (custom binaries, if applicable)
/opt/ycelib - 500 MB (supporting libraries)
/var/opt/yce - 3 to 15 GB (logs and working data)
/var/opt/shared - 6 to 15 GB (tftp, os-files)
/var/opt/mysql -  4 to 10 GB (mysql data)
```

Alternative filesystem assignments can be used where deemed useful to the local administration. It is advisable tough to assign /var/opt/mysql, /var/opt/shared, and var/opt/yce to their own filesystems to limit the impact of overflowing filesystems.

Q&A on NetYCE server setup

System Users

NetYCE uses a local (unix) user account, 'yce', that owns and runs all processes and files. The 'yce' user is a functional user and should not have a direct login. The account is used by administrators to perform maintenance so 'sudo' access to the 'yce' account is required.

The 'yce' user characteristics:

- local functional user, non-login
- bash-shell
- no password ageing
- belongs to group id 'nms'
- requires crontab configuration
- must have outgoing mail privileges for operator-signalling (domain selection determined by customer)
- localhost incoming mail privileges (process monitoring also uses local mail for signalling)
- the standard 'ping' and NetYCE 'fping' should be executable (suid root)
- requires some 'sudo' permissions without password (process restarts mostly)

Most frequent maintenance or setup tasks are available using the web-GUI (process monitoring and restarts, update installation, wiki updates, license file updates, common configuration files).

However, for application maintenance purposes, access to the 'yce' shell should be available through generic user-accounts. A 'su' or 'sudo su' should be setup to achieve this. The 'yce' account is mostly used to update configuration files or execute scripts to achieve this. Their use is infrequent tough. Also examination of the many detailed log files is a task often executed through this account.

A second user account is required for the secure transfer of (configuration and os) files between the devices and the NetYCE server(s). Traditionally only TFTP was used, but this increasingly archaic

protocol is found to be too limited in security and performance to be acceptable in modern networks. NetYCE supports SFTP and FTP as well, but its use requires a secondary functional user named 'ycicle' that shares the file permissions with 'yce' but is allowed a SFTP / FTP restricted login from the managed devices using a password. For security reasons, the 'ycicle' user is locked in the file-sharing directory tree (chroot /var/opt/shared) and its password is **only** available to the NetYCE automated tasks.

Details on this setup are found in FTP and SFTP setup

Root-permissions

At the initial installation of NetYCE, root permissions are required. For non-major updates and day-to-day maintenance some specific 'sudo' permissions to restart a few daemon processes are sufficient. The NetYCE application does not use 'root' permissions anywhere. The 'root' password is fully owned bu the customer.

By default only the NetYCE process manager (yce_psmon) requires root startup permissions. The file-transfer protocols that bind to well-known-ports (httpd, tftp, ftp, sftp) also demand to be started by a 'root'-owned process. This task is (amongst others) handled by yce_psmon. Its primary task is starting, stopping and restarting of the NetYCE processes and includes killing off misbehaving parts (exceeding preset cpu, memory or thread limits).

By preference the SElinux is disabled or used in 'permissive' mode. Using it in 'enforcing' mode is possible, but its configuration proved to be a 'pain-in-the-but' Especially for the interaction between the Web-GUI and the many back-end tasks proved hard to setup properly.

However for the servers where 'enforcing' is a must, NetYCE supports a mode where the httpd daemon is allowed to run with 'non-root' permissions. This situation simplifies the SElinux setup somewhat and reduces the security risks that are involved with the default httpd 'root' owned setup. This mode uses `setcap cap_net_bind_service=+ep /usr/sbin/httpd` which must be re-issued by a 'root' user after every httpd version update. (also see https://wiki.apache.org/httpd/NonRootPortBinding)

This mode can be selected during the NetYCE installation. The SElinux setup and maintenance is not a responsibility of NetYCE.

During the initial install 'root' permissions are needed to:

- creating the 'yce' user and the 'nms' group including the home directory
- configuring the 'sudo' permissions for the 'yce' user
- setting SElinux to 'permissive' if it was 'enforcing'
- configuring networking such that `hostname -s, -d, and -i` responses match correct values
- disabling iptables configuration
- updating startup configuration using `chkconfig` for httpd en yce_psmon daemons
- installation of rpm packages 'fping' and 'vsftpd'
- assigning suid to fping
- installation of the rpm packages for MariaDB
- creating the initial net_setup.xml en yce_setup.xml using NetYCE install scripts
- initial copying of netYCE license file
- initial installation of the NetYCE 'YcePerl' distribution (system Perl unaffected)

- initial installation of the NetYCE distribution package
- MariaDB configuration file owner change to yce.nms
- MariaDB data directory owner change to yce.nms
- Apache httpd configuration file owner change to yce.nms
- Apache httpd log-file owner change to yce.nms
- installation of `mod_ssl` package voor httpd (for https)
- installation of https certificate (for https)
- installation of `mod_php` package voor httpd (for local Wiki)
- creation of user ‘ycicle:nms’ (for sftp file-transfers)
- configuration of sftp with chroot for ‘ycicle’ (for sftp)
- configuration of `vsftp` with chroot for 'ycicle' (for ftp)

NetYCE will issue various packages for the installation on RedHat/CentOS RHEL 6.x X86_64:

- NetYCE license file
- NetYCE YcePerl installable file
- NetYCE distribution installable file
- Default NetYCE sample database

Installation files to install the fping, vsftpd, and MariaDB packages if desired.

MariaDB

MariaDB can be downloaded for updates by the customer (See Installation on RedHat Linux, MariaDB). This MySQL-base database will run using 'yce' permissions. The log files are therefore accessible for the 'yce' user.

Any 'mysql' user created during the installation of MariaDB can safely be removed afterwards.

Directories

NetYCE installs her distribution in various directory trees using `/opt'; as base. All variable data is located in directory trees using` /var/opt'' as base:

```
/opt/yce        - netYCE distribution tree
/opt/ycelib     - netYCE perl en wiki distribution tree
/opt/nms        - customer-specific integrations
/var/opt/yce    - netYCE job and log data
/var/opt/mysql  - database data
/var/opt/shared - tftp, sftp and shared data
```

Log files

NetYCE maintains a wide variation of log files for its various processes and daemons. All are located in the '/var/opt/yce/logs' directory and all are self-rotating.

Executed change jobs create their own set of temporary data and log files. They are all located in the /var/opt/yce/jobs' directory tree and organized by Job-id. This history is also self-maintaining.

In the database an series of logs are maintained that register all user actions and their effects, the jobs created and their effects, the configuration changes observed, and the API-calls processed. By default these detailed logs are preserved for 400 days for traceability reasons. These logs too are maintained in self-rotating files to allow external processing.

The unix system log-files are not used in any way.

Syslog usage

Syslog reporting is currently only used by the NetYCE process monitoring daemon. It reports its actions using log-files, syslog and email notifications.

Crontab dependencies

NetYCE servers rely on the crontab functionality for several of its capabilities. These include:

- scheduled reports from users
- daily database maintenance (autorepair errors and fragmentation, logging tables)
- daily job details and log maintenance
- daily database backup and history
- daily maintenance of tftp/sftp/shared files

All crontab entries strictly use the 'yce' user tab. No NetYCE crontab entries exist under 'root'.

Monitoring

The NetYCE process monitor, 'yce_psmon', is the primary application monitor. This daemon ensures all required processes are running properly. It will monitor memory and cpu usage and restarts those processes that misbehave. It also ensures the number of childs of a process do not exceed preset limits.

Another daemon monitors the database replication and controls database switchovers. It is also tasked with license monitoring and database column encryption.

Connection matrix

For the information on the communication protocols used between the various NetYCE components and the networking devices, the users and NMS infrastructure, see YCE Connection matrix

From:
https://wiki.netyce.com/ - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php?id=maintenance:general:system_requirements

Last update: **2024/07/03 12:31**