NoMachine Workstation - Installation and Configuration Guide

Table of Contents

Introduction

How to set-up the Workstation

Connect to the Workstation

Configurations and Optimizations

Administration of Workstation

Federating the Workstation Under a Cloud Server

Introduction

1. NoMachine Workstation Installation and Configuration Guide

Welcome to the NoMachine Workstation - Installation and Configuration Guide v. 7.

What is NoMachine Workstation for?
NoMachine Workstation is a standalone server for Linux that allows up to four concurrent virtual desktops to run on the host where it is installed. Designed to offer individual instances of the remote desktop (terminal services), it provides users with their own separate desktop environment. With the Workstation each user has his/her own desktop or application, and can store, manage files inside the session and even share his/her own resources with another user.

Available for Linux, the Workstation accepts connections via a browser (thanks to its built-in web server) or via the NoMachine client.

Additionally, it can also be federated under a Cloud Server. This solution is suitable to centralize access to multiple NoMachine servers distributed across a LAN or WAN environment.

A Graphical Interface
The NoMachine Workstation server package includes a User Interface (UI) for administering the server and its services (Server settings).

Most common operations detailed in this guide can be performed by the NoMachine UI and the Server settings panel running on the local installation of the server.

More details about the Server UI can be found in the dedicated guide available in the Documents section at: https://www.nomachine.com/all-documents

The Workstation comes also with a client UI for running sessions and connecting to other remote desktops.

The server is fully operative once installed
Installation provides a fully operative NoMachine server with a default configuration suitable for the majority of environments. All the necessary services are automatically started.

A standalone server
NoMachine Workstation is a single server (standalone server) to all effects. NoMachine Workstation supports up to four concurrent virtual desktops and/or single applications ('custom sessions'). A virtual desktop is a personal instance of the remote Linux desktop. Sharing of a virtual desktop is also supported. The number of users is not limited.

A federated server
NoMachine Workstation can be also federated under a Cloud Server v. 7 which provides a single point of access to multiple server subsystems. In this case, it's possible to configure the Workstation to not accept direct connections. For more specific instructions about federating the Workstation, refer to the Cloud Server administrative's guide available in the 'Installation' section at https://www.nomachine.com/all-documents.

1.1. About This Guide

Document Conventions and Important Notices
The following conventions are used in this Guide:

BaseDirectory
is the base directory where the NoMachine binaries and libraries are installed.
By default, BaseDirectory is: /usr.

InstallationDirectory
is: *BaseDirectory*/NX, i.e. **/usr/NX**.

The command line interface
NoMachine server and node programs have a command line interface to execute operations.

You need to be a privileged system user to access all these functionalities. These commands can be run from an xterm or similar using the sudo utility or as root.

On Linux invoke the 'nxserver' and 'nxnode' programs from /etc/NX, for example:

```
$ sudo /etc/NX/nxserver --status

Printing the server and node usage doesn't require to be a privileged user, instead:

$ /etc/NX/nxserver --help

$ /etc/NX/nxnode --help

The 'nxserver --help' and 'nxnode --help' display the list of all the available commands and options and their description.
```

Online Resources
Visit the NoMachine Support Area to access a variety of online resources including the NoMachine Forums, tutorials and FAQs: https://www.nomachine.com/support. Or log in to your customer area to open a support enquiry.

Find a list of all documents and tutorials: https://www.nomachine.com/all-documents

Use the Knowledge Base search engine to access articles, FAQs and self-help information: https://www.nomachine.com/knowledge-base

Leave Feedback About This Guide
Our goal is to provide comprehensive and clear documentation for all NoMachine products. If you would like to send us your comments and suggestions, you can use the contact tool available at https://www.nomachine.com/contact-request, selecting Web Quality Feedback as your option.

2. Install the Workstation

2.1. Prerequisites

Supported Operating Systems
Linux 32-bit and 64-bit

RHEL 4 to RHEL 8
SLED 10 to SLED 15
SLES 10 to SLES 15
openSUSE 10.x to openSUSE 15.x
Mandriva 2009 to Mandriva 2011
Fedora 10 to Fedora 31
Debian 4.0 to Debian 10
Ubuntu 8.04 to Ubuntu 20.04

Raspberry Pi 2/3/4 ARMv6/ARMv7/ARMv8

Hardware requirements
Intel Core2 Duo or AMD Athlon Dual-Core or equivalent
1 GB RAM
Network connection (either a LAN, or Internet link: broadband, cable, DSL, etc...)
Size required on disk:
Linux 195 MB

Software requirements
A desktop environment must already be installed. This applies also to headless Linux machines.
Connections by the web and by NoMachine clients are supported.

Compatibility with older versions
Even if it's advisable to upgrade client installations to the same version 7 of the Workstation, compatibility with clients v. 6 and 5 is preserved.
Note also that when the Workstation works as a federated server, NoMachine Cloud Server v. 7 requires a client v. 7 or 6.

2.2. Linux Installations

Installing for the first time
You can install, update and uninstall using the graphical package manager of your Linux distribution or from command line by running commands from an xterm or similar with the sudo utility, or as root user if you don't have sudo installed. Instructions below refer to installation by command line.

If you own a **customer license** we recommend to download the package from your Customer Area: https://www.nomachine.com/support#login.

Successive updates
The update procedure for server and node installations requires all NoMachine services to be stopped in order to correctly replace libraries and binaries. This implies that the Workstation is not accessible to users during the update procedure. Current sessions will be terminated, users will be able to

connect again later.

There are two ways to update your current installation:

I Automatic updates

You can update your installation from our repositories. Just run the NoMachine User Interface from your Programs Menu and open the Server -> Settings -> Updates panel and click on the 'Check now' button.

NoMachine has the automatic check for updates enabled: it will check the repositories every two days to verify if updates are available. In this case, the server will prompt a dialog informing that a new version is available but **it will never automatically update the current installation**.

Checking for updates can be disabled from that dialog by selecting the 'Don't ask again for this version' option or in the Updates panel by unchecking the 'Automatically check for updates' option.

Detailed instructions for configuring the Automatic Updates are available in the Documents section on the NoMachine web site: https://www.nomachine.com/all-documents .

II Update with NoMachine packages

Alternatively, download the latest available package from the NoMachine web site and click on the executable file to launch Setup. As for the installation, Setup will guide you through all steps necessary for updating your installation.

If you own a **customer license** we recommend you download the package from your Customer Area: https://www.nomachine.com/support#login.

2.3. RPM Packages

If you want to **install to default location**, namely /usr/NX/:

INSTALL

```
# rpm -ivh <pkgName>_<pkgVersion>_<arch>.rpm

To find out which NoMachine package you have installed (you will get the full name of the package), run:

# rpm -qa | grep nomachine
```

UPDATE

```
# rpm -Uvh <pkgName>_<pkgVersion>_<arch>.rpm
```

UNINSTALL

```
# rpm -e nomachine-workstation
```

TIP

For **non-default locations**, for example /opt/NX:

INSTALL

```
# rpm -ivh <pkgName>_<pkgVersion>_<arch>.rpm --prefix /opt
```

UPDATE

```
# rpm -Uvh <pkgName>_<pkgVersion>_<arch>.rpm --prefix /opt
```

UNINSTALL

```
# rpm -e nomachine-workstation
```

2.4. DEB Packages

If you want to **install to default location**, namely /usr/NX/:

INSTALL

```
$ sudo dpkg -i <pkgName>_<pkgVersion>_<arch>.deb
```

To find out which NoMachine package you have installed (you will get the full name of the package), run:

```
$ dpkg -l | grep nomachine
```

UPDATE

```
$ sudo dpkg -i <pkgName>_<pkgVersion>_<arch>.deb
```

UNINSTALL

```
$ sudo dpkg -r nomachine-workstation
```

TIP

For **non-default locations**, for example /opt/NX:

INSTALL

```
$ sudo NX_INSTALL_PREFIX=/opt dpkg -i <pkgName>_<pkgVersion>_<arch>.deb
```

```
UPDATE

$ sudo NX_INSTALL_PREFIX=/opt dpkg -i <pkgName>_<pkgVersion>_<arch>.deb

UNINSTALL

$ sudo dpkg -r nomachine-workstation
```

2.5. TAR.GZ Packages

If you want to **install to default location**, namely /usr/NX/, ensure that package is placed there.

INSTALL

```
$ cd /usr

$ sudo tar xvzf <pkgName>_<pkgVersion>_<arch>.tar.gz

$ sudo /usr/NX/nxserver --install
```

UPDATE

```
$ cd /usr

$ sudo tar xvzf <pkgName>_<pkgVersion>_<arch>.tar.gz

$ sudo /usr/NX/nxserver --update
```

UNINSTALL

```
$ sudo /usr/NX/scripts/setup/nxserver --uninstall

$ sudo rm -rf /usr/NX
```

TIP

For **non-default locations**, for example /opt/NX:

```
INSTALL

$ sudo NX_INSTALL_PREFIX=/opt /usr/NX/nxserver --install

UPDATE

$ sudo NX_INSTALL_PREFIX=/opt /usr/NX/nxserver --update
```

UNINSTALL

```
$ sudo /opt/NX/scripts/setup/nxserver --uninstall
$ sudo rm -rf /opt/NX
```

2.6. Activating the License (for Customers)

Customer packages
include a temporary (30-days) node.lic and server.lic files for evaluation.
Such license files have to be replaced with the customer's license files acquired from NoMachine.via the NoMachine UI in Settings -> Server -> Updates panel: click on 'Server subscription' and 'Node subscription' to read the current license file. Click on 'Replace' button to activate a new license file.

TIP

Starting from version 7, the server will cease to work once the license is expired. Please contact your NoMachine provider or the Support Team for renewal options.

To verify from command line that server.lic and node.lic are correctly in place and check their validity, you can run:

```
$ sudo /etc/NX/nxserver --subscription

$ sudo /etc/NX/nxnode --subscription

$ sudo /etc/NX/nxserver --version

$ sudo /etc/NX/nxnode --version
```

3. Initiating a NoMachine Connection (end-user's side)

Pre-requisite: you need to have a valid account on the Workstation host, password cannot be empty. Your account can be a local account or a LDAP account or an AD account. NoMachine doesn't check if the source of users' account information is for example LDAP or local account database. Be sure to configure the login in advance.

3.1. Connecting by Browsers Via Workstation Web Tools

Once installation is complete, Workstation is ready to go.

The end-user should point the browser on his/her device to:

http://SERVER:4080

Where SERVER is either the name or IP address of the host you want to reach.

E.g., if Workstation is installed on a host named 'myserver.com', the URL will look like this:

http://myserver.com:4080

The connection will be automatically switched to HTTPS protocol:

https://myserver.com:4443/nxwebplayer

In the login form, the end-user has to provide username and password of his/her system account on the Workstation host and connect.

TIPS

I Auto-reconnection is supported: when the connection is lost for whatever reason (including when browser's computer has entered sleep mode), the NoMachine web application will automatically try to reconnect for as long as the user keeps the web page open. If reconnecting is not possible, then the user will have to reconnect manually.

II IPv6 is supported: specify the IP address of the server host in IPv6 format (e.g. 2001:0:5ef5:79fb:30c6:1516:3ca1:5695) if you want to use it instead of IPv4.

You can find a step-by-step tutorial for connections by the web in the 'Tutorials' section at: https://www.nomachine.com/all-documents

3.2. Connecting by NoMachine Client

Install NoMachine Enterprise Client on your device. You can use also NoMachine free or whichever other server type, they all provide the client UI for connecting to remote computers.

You can find a step-by-step tutorial for creating virtual desktops the 'Tutorials' section at: https://www.nomachine.com/all-documents. If you're using the Enterprise Client, please see also its Installation and Configuration Guide in the 'Installation' section at the same web page above.

TIPS

I Auto-reconnection is supported: when the connection is lost for whatever reason (including when the client host has entered sleep mode), the client will automatically try to reconnect for as long as the user keeps the User Interface open. If reconnecting is not possible, then the user will have to reconnect manually.

> II IPv6 is supported: specify the IP address of the server host in IPv6 format (e.g. 2001:0:5ef5:79fb:30c6:1516:3ca1:5695) if you want to use it instead of IPv4.

3.3. Preventing Users from Storing their Credentials

To prevent NoMachine users from storing their credentials, use the EnableCredentialsStoring key in the server configuration file. The accepted values are:
player Only users connected via NoMachine client are enabled to save username and password in the connection file stored on their devices (computer, tablet etc ...)
webplayer Only users connected via browser can choose to save their access credentials. They are stored in the browser's cookie, given that the user's browser has cookies enabled.
both All users, regardless if connected via NoMachine client or via web, can store their credentials.
none Users cannot save their username and password. They will be requested to provide their log-in credentials at each connection.

For example, to allow only users connecting via NoMachine client to store credentials, set in the server configuration file:

```
EnableCredentialsStoring player
```

4. Configuring NoMachine Workstation

4.1. Configuring Web Sessions

The configuration file for the web player program (which provides the graphical front-end) and the web client program (which manages web sessions) is server.cfg, located in the BaseDirectory/NX/etc directory: /usr/NX/etc/server.cfg.

Default settings
The Section directive defines settings for the NoMachine server(s) where the web player application will connect. This directive, by default, points to localhost and looks like:

```
Section "Server"

Name "Connection to localhost"
Host localhost
Protocol NX
Port 4000
```

```
Authentication password
```

```
EndSection
```

Name is a label that can be displayed as an alternative to show hostname of the server. To change the name to be displayed, it's necessary to enable also the EnableWebConnectionName key in server.cfg: `EnableWebConnectionName 1`

Host is IP or hostname of the NoMachine server host.
If different than localhost, it's then necessary to update manually the list of known hosts as explained here: https://www.nomachine.com/AR06P00984

Protocol specifies the protocol, NX or SSH, that the web player will use to connect to the NoMachine server. Default is:

```
Protocol NX
Port 4000
```

To use SSH protocol with default port, set for Linux and Mac:

```
Protocol system
Port 22
```

and for Windows:

```
Protocol system
Port 4022
```

Port indicates the listen port for the server, by default 4000 for connections by NX protocol, 22 for connections by SSH protocol on Linux and macOS and 4022 for connections by SSH on Windows.

This setting has to be changed only when NoMachine is configured to listen on a non-default port. Changing the port for web connections requires a manual procedure available here: https://www.nomachine.com/AR06N00888

Authentication defines the authentication method to be used when connecting by the web, by default 'password':

```
Protocol NX
Authentication password
```

and:

```
Protocol SSH
Authentication password
```

Key-based authentication is supported at the moment only for NX protocol, instructions to set it up are available here: https://www.nomachine.com/AR03Q01020

TIP

NoMachine uses by default port 22 for SSH protocol on Linux.The default port for NX protocol is 4000. In order to change the port for NX protocol, change the port for the nxd service and restart it. See the paragraph **'Connecting by NX Protocol'** . To change the port for connections by SSH it's necessary to modify the listen port for the SSH server on the system.

Other settings in server.cfg specific for web sessions are qualified by the **'Web'** keyboard in their name.

Showing hostname or a custom label
To display hostname or IP of the Workstation host when connecting by the web, set the following key. This is the default:

```
EnableWebConnectionName 0
```

To display the label set in the 'Name' field of Section "Server" set:

```
EnableWebConnectionName 1
```

Hiding the tutorial wizard at web session startup
To not display the tutorial wizard for the menu panel at session startup, set:

```
EnableWebMenuTutorial 0
```

and to show it (this is the default):

```
EnableWebMenuTutorial 1
```

4.2. Managing Workstation Web Services

You can start and stop the NoMachine HTTP server (nxhtd)
from the Server preferences User Interface -> Server preferences -> Network services panel. From the NoMachine User Interface you can also change the port where the web server will be listening (by default 4080 and 4443 for secure connections).

From command line it's possible to do the following.

Stop, start or restart nxhtd

```
/etc/NX/nxserver --stop nxhtd

or:

/etc/NX/nxserver --start nxhtd

or:

/etc/NX/nxserver --restart nxhtd
```

Automatic restart of the nxhtd service
Each service is automatically restarted at the next boot. You can change the default behavior for the nxhtd service by setting:

```
/etc/NX/nxserver --startmode nxhtd manual
```

```
or to enable the automatic restart of the service:

/etc/NX/nxserver --startmode nxhtd automatic
```

The nxserver --startmode command operates on the StartHTTPDaemon server configuration key:

StartHTTPDaemon Automatic

and:

StartHTTPDaemon Manual

Disabling the starting of the NoMachine HTTP server
Edit the server configuration file and remove HTTP from the ClientConnectionMethods key. It should then look like:
ClientConnectionMethods NX,SSH

Then restart NoMachine server to make this change effective:

```
/etc/NX/nxserver --restart
```

4.3. Using an Alternative Apache Web Server

NoMachine Workstation is designed to provide a fully integrated service to deploy sessions on the web which doesn't require additional software to be installed or manual configuration. The minimal Apache web server, nxhtd, provides the necessary modules and is pre-configured to work with the web player application.

However, it is possible to run the web player application with an alternative Apache web server. Look for detailed instructions in our Knowledge Base, section Articles, by searching for the 'Apache' keyword: https://www.nomachine.com/knowledge-base.

4.4. Web Optimizations: Using WebRTC (Real-Time Web Communication)

The implementation of WebRTC support in browser-based remote desktop sessions has inititally been released as beta and must be enabled explicitly by the administrator by editing the server.cfg file.

With the help of a STUN/TURN server for negotiating NAT traversal, peer-to-peer WebRTC communication can be established also when the web session has to be run behind a NAT.

STEP 1: Uncomment and set the AcceptedWebMethods key as follows to enable the use of WebRTC:

```
AcceptedWebMethods webrtc
```

STEP 2: If the node machine where the web session will be started is behind a NAT, you need to uncomment the following section in server.cfg:

```
Section "STUN"

Host hostname
Port portnumber
User username
Password password

EndSection
```

and provide relevant information to contact a STUN or TURN server. In this last case change Section name to "TURN".

Specific articles can be found in the Knwoledge Base,https://www.nomachine.com/knowledge-base.

5. Compression Techniques and Optimizations

5.1. Video Stream Encoding in Web Sessions

In the case of web sessions, by default, session data is streamed in video frames compressed and decompressed by using the MJPEG lossy algorithm, which is the video-format widely supported by browsers.

Oher video codecs like VP8 and H.264, require a browser which supports WebRTC and HTML5.

NoMachine web sessions use the H.264 video streaming when the following requirements are all met, otherwise VP8 is used. In practice, when WebRTC is enabled, the H.264 or VP8 encoding will be used, otherwise MJPEG will be used:

I WebRTC is enabled.
II The browser supports WebRTC and the H.264 decoding

In practice, when WebRTC is enabled, the H.264 or VP8 encoding will be used (if the browser supports WebRTC), otherwise the classic web media exchange protocol will be used and MJPEG will be the codec.

TIP

H.264 hardware encoding is possible when the Workstation host machine has an hardware accelerated video cards (GPU) with Nvidia Kepler microarchitecture onward or Intel Quick Sync

processors or AMD card (at the moment only on Windows).
Enabling HW acceleration by Quick Sync requires however a manual configuration as explained here: https://www.nomachine.com/AR09O00938

Optimizations
Optimizations can be done in two ways: (I) by adjusting display settings in the session or (II) by enabling WebRTC.

I **Adjusting display settings in the web sessions**
To access NoMachine display settings, open the NoMachine menu inside the web session: press ctrl+alt+0 or click on the page peel in the upper right corner of the window to open it. Then click on the 'Display' button and finally on 'Display settings'. From this panel you can do the following.

Change the display image quality
Increasing the quality will mean to apply a minor compression ratio, the image will be clearer, but more bandwidth will be used.

Disable network-adaptive display quality
This will anchor the display quality to the fixed value specified in the Display quality slider, making it independent from the current network congestion. This is not recommended when there is a very limited bandwidth.

Disable multi-pass display encoding
Default settings within the encoding will work to refine the image progressively to the target quality (as specified in the Display quality slider) starting from a lower quality version of the image during moments of inactivity of the desktop. Disabling this refinement sends the image directly with target quality. Not recommended when there is limited bandwidth.

II **Enabling WebRTC**
NoMachine web sessions use by default the classic web media exchange protocol for the two-way browser/web server communication. WebRTC (Real-Time Web Communication) is also supported and can be enabled as explained in the next paragraph.
Enabling WebRTC allows to use the H.264 video streaming (when possible) or VP8 which optimize the user experience with multimedia applications and content.

TIP

You may verify which encoding method is in use from the NoMachine menu inside the session: press ctrl+alt+0 or click on the page peel in the upper right corner of the window to open it. Then click on the 'Display' button and finally on 'Display settings'. The codec actually in use is reported at the bottom left of the menu.

5.2. Video Streaming Encoding in Client Sessions

Sessions run by NoMachine client use a combination of video and image encoding based on standard codecs and a number of techniques developed by NoMachine. Frames are encoded into a video stream optimized by means of a compression and decompression algorithm of real-time image and audio data. VP8, H.264 and MJPEG encoding are supported.

In general VP8 and H.264 are suitable for all situations, while MJPEG can be an alternative when the end-user's computer is less powerful and the user is experiencing slow responsiveness.

The display encoder can be changed on the server:

from the User Interface
In the User Interface in the Server Performance panel.

or in the **node configuration file**
Enable the use of a specific codec by editing the node configuration file and enabling the following two keys:

```
EnableDisplayServerVideoCodec 1
DisplayServerVideoCodec CODEC
```

where CODEC can be: 'vp8','h264' or 'mjpeg'. For example:

```
EnableDisplayServerVideoCodec 1
DisplayServerVideoCodec mjpeg
```

5.3. The X11 Vector Graphics Mode in Client Sessions

The X11 vector graphics mode (previously called 'lightweight mode') is enabled by default for (i) virtual desktops and (ii) custom sessions in floating window mode. This mode is mainly a set of NoMachine techniques to compress and optimize the X11 protocol (by applying the same algorithms available with the NX compression protocol v. 3). These compression techniques are applied to all non-video content like textual elements, while multimedia content is encoded in a video stream (VP8 or H.264).

The X11 vector graphics mode is useful for avoiding loss of image quality and in general is the best option when working with traditional User Interfaces or a large amount of text. However it's not advisable for multimedia content or applications with many graphical effects.

This mode can also help to reduce bandwidth usage, decrease the HW requirements on client and server (expensive video encoding/decoding operations are applied only to multimedia content), increase responsiveness on slow link and end-users' clients without hardware accelerated video encoding/decoding capabilities.

You can disable/enable the X11 vector graphics mode
via the User Interface
in the Settings -> Server -> Performances panel

or **in the node configuration file**
Edit the node configuration file, uncomment and set the AgentX11VectorGraphics key (previously named 'AgentLightweightMode') to '0' for disabling the the X11 vector graphics mode:

```
AgentX11VectorGraphics 0
```

or to enable it:

```
AgentX11VectorGraphics 1
```

TIPS

I In the case of slow bandwidth, decreasing the quality level of images could help but if you need to have a perfect image without quality loss, you have to increase the display quality instead. It's also suggested to disable multi-pass encoding to avoid the 'out of focus' effect: multi-pass is an encoding technique which uses multiple passes to progressively reach the best definition of the image.

II Quality level and multi-pass encoding can be tuned from the NoMachine menu inside the session in the Display -> Change settings panel. (Ctrl+alt+0 or click on the right upper corner of the window to open the NoMachine menu).

5.4. Supporting OpenGL Applications in Virtual Sessions

In NoMachine virtual desktops and custom sessions, OpenGL rendering is done by default by software components. This means that rendering tasks are accomplished by CPU and not offloaded onto the GPU. Such operations canbe resource-demanding, especially in the case of 3D desktop graphics effects, and make the user interface look slow.

A possible alternative is to configure the NoMachine server to use the VirtualGL libraries (included in the NoMachine package) and therefore activate support for HW accelerated OpenGL applications. This allows OpenGL applications, namely 3D applications, to use server side graphics hardware.

In order to activate support for VirtualGL, follow instructions at
http://www.nomachine.com/AR05P00982

6. Workstation Configuration

6.1. Configuration Files

The configuration file for the nxserver and nxweplayer/nxwebclient programs is server.cfg. The configuration file for the nxnode program is node.cfg.

They are placed in (default installation):
/usr/NX/etc/server.cfg
/usr/NX/etc/node.cfg

The Default Configuration
NoMachine Workstation comes with a default configuration that grants a working setup for the majority of environments. NoMachine administrators can tune their installation at any moment and according to their specific needs by setting the related configuration keys. In some cases this will require to restart all NoMachine services.

Edit the Configuration Files
NoMachine configuration files are text files made up of a number of key-value pairs. All the configuration files can be edited manually by a text editor, e.g. 'vi'.

Be sure to uncomment the configuration key (i.e., remove the '#' pre-pended to the key) to set a value different from the default.

When a configuration key supports an on/off status, set value to '0' to disable it and to '1' to enable it.

Make Changes to the Default Configuration Effective
Changes will be effective with the next new connection without the need to restart the server if not otherwise specified.

7. Services Management

Installation and upgrade procedures take care of configuring and starting all the necessary services to make NoMachine Workstation ready to accept and serve user requests for virtual desktops and custom sessions. The necessary services are configured to be restarted at each reboot of the host machine.

7.1. Accepting Connections

When you are sit in front of the computer where the Workstation is installed or you're connected there by NoMachine, you can switch off/on the ability to accept connections to your desktop via NoMachine. When you disable the sharing of your screen, nobody can connect.

You can configure this setting via the NoMachine Monitor menu (right click on the !M icon in the system tray to open it) by clicking on item "Accept connections".

This setting lasts until you change it again, even when you physically log-out from the system.

See also this tutorial for more details: https://www.nomachine.com/disabling-access-to-your-local-desktop

TIPS

I Be careful if you decide to disable accepting connections when you are connected from remote: you will be no longer able to reconnect to the desktop via NoMachine once the current session is closed. In this case, you can recover the ability to connect via NoMachine by changing settings in the Monitor menu on the physical computer.

II It's posible to hide the 'Accept connections' item from the !M menu by configuring the NoMachine node.cfg file on that computer and setting:

```
EnableAcceptingConnections 0
```

Be sure to remove the pre-pending # from the key name.

7.2. Stopping and Starting Workstation (nxserver) and Services

All NoMachine services can be stopped via:

the User Interface
all NoMachine services can be stopped by the Server status User Interface ('Shutdown the server'). When doing so, you will be asked if services must be started at the next reboot or not. You can restart services also from the Server status User Interface ('Start the server').

or **from command line**.

Stopping all the NoMachine services

```
/etc/NX/nxserver --shutdown
```

This will completely disable access to the server host machine and terminate all sessions running on that host. By default, all services will be restarted when the machine is booted. To override this behavior, specify the --startmode option when stopping the services:

```
/etc/NX/nxserver --shutdown --startmode manual
```

Starting NoMachine server and services

```
/etc/NX/nxserver --startup
```

All services will be restarted at the next reboot. To not start services when the machine is rebooted, specify the start mode while running the --startup command:

```
/etc/NX/nxserver --startup --startmode manual
```

Specifying the start mode
It's possible to set the 'start mode' (if services will be started automatically at boot or not) by using:

```
/etc/NX/nxserver --startmode manual
```

or:

```
/etc/NX/nxserver --startmode automatic
```

Stopping and restarting NoMachine server and services

```
/etc/NX/nxserver --restart
```

7.3. Stopping and Starting a Network Service

The NoMachine network services available for NoMachine Workstation are nxd and nxhtd:

Program	Default port	Scope
nxd	4000	Accept connections by NX protocol
nxhtd	4080 and 4443	Accept connections by HTTP protocol (connections by the web)

You can stop a single service:

via the User Interface
in the Settings -> Server -> Ports services panel. You can also choose the start mode: whether the service has to be started automatically at the next boot or not.

or **from command line**.

Stopping a service

```
/etc/NX/nxserver --stop SERVICE
```

where SERVICE can be:
nxd, the Network Server for accepting connections by NX protocol
nxhtd, the NoMachine web server for web sessions

Starting or restarting a service

```
/etc/NX/nxserver --start nxd

/etc/NX/nxserver --start nxhtd

or:

/etc/NX/nxserver --restart nxd

/etc/NX/nxserver --restart nxhtd
```

Specifying the start mode
By default each service is automatically restarted at the next boot. You can configure that on a per-service basis by running:

```
/etc/NX/nxserver --startmode SERVICE manual

or to restore the default behavior:

/etc/NX/nxserver --startmode SERVICE automatic
```

Commands above operate on the configuration keys listed below. You can change them manually in the server configuration.

Configuration	Key setting
Enable automatic starting of the NX Network server, nxd	StartNXDaemon Automatic
Disable automatic starting of the NX Network server, nxd	StartNXDaemon Manual
Enable automatic starting of the NoMachine web server, nxhtd	StartHTTPDaemon Automatic
Disable automatic starting of the NoMachine web server, nxhtd	StartHTTPDaemon Manual

7.4. Local and Network Ports

For each session, NoMachine uses ports that are used only locally on the server host and network ports.

Some ports are mandatory and must be free, e.g. the session display number and the connection port. Other ports are used for services that can be disabled (e.g. USB forwarding, UDP communication).

Local port	Description	How to change the default
11000 + DisplayBase	Session display. If this port is already in use, NoMachine will look for a free port by incrementing DisplayBase up to the value set in the DisplayLimit server configuration key.	DisplayBase (by default 1001) and DisplayLimit (200) are defined in server.cfg
20000	Communication port between the session's nxserver process and the main server process.	Add the ServerSlaveBase key to the end of server.cfg and specify a value
24000 + DisplayBase	Session's monitor port.	DisplayBase (by default 1001) and DisplayLimit (200) are defined in server.cfg
5473 and 5483	USB devices forwarding.	Disable USB sharing by setting EnableUSBSharing none in node.cfg

Network port	Description	How to change the default
	TCP port for the NoMachine display	

6000 + DisplayBase	service. If this port is already in use, NoMachine will look for a free port by incrementing DisplayBase up to the value set in the DisplayLimit server configuration key.	DisplayBase (by default 1001) and DisplayLimit (200) are defined in server.cfg.
5353	UDP port for the MDNS service to broadcast computer's information over the LAN.	Disable the service by setting EnableNetworkBroadcast 0 in server.cfg.
4000	TCP port for the NoMachine Network service (nxd) and connections via NX protocol. This port must be open in the firewall and mapped to the external IP of the server host.	Set NXPort in server.cfg and restart the nxd service.
4011 - 4999	UDP port range.	Set UDPPort in server.cfg to define a different range. UDP can be disabled on client side.
22	TCP port for connections via SSH protocol. This port must be open in the firewall and mapped to the external IP of the server host.	Set a different port for the system SSH server and align value set for SSHDPort in server.cfg. Then restart the NoMachine server.
4080 and 4443	HTTP and HTTPS port for web connections. These ports must be open in the firewall and mapped to the external IP of the server host.	Change 'Listen' directives in htd.cfg and restart the nxhtd service.
20000 - 30000	External ports range for UPnP port mapping.	Set NXUPnPPort in server.cfg to define a different range. Set EnableUPnP none in server.cfg to disable port mapping.
5040 + x	Port opened between client and server for each USB device. Port number is defined by 5040 + x where 'x' is the first free port retrieved starting from port number 5040.	N/A
4000	Automatic updates from NoMachine repositories.	Updates are managed by nxd. Disable automatic updates by setting UpdateFrequency 0 in server.cfg.
5473 and 5483	USB devices forwarding.	Disable USB sharing by setting EnableUSBSharing none in node.cfg.

7.5. Hiding the NoMachine Monitor and Notification Messages

It is possible to hide or show the !M (the Monitor) icon in the system tray. When the icon is hidden, notification messages will still be displayed when users are connecting. This can be configured:
from the User Interface:
In the Settings -> Server > Security panel, check 'Hide the NoMachine icon in the system tray'. When the icon is hidden, notification messages will still be displayed when users are connecting.

or in the **node configuration file**

This setting is ruled by the DisplayMonitorIcon key in the node configuration file. If you change them manually by editing the file, you then need to restart the server to make changes effective.

To hide the !M in the system tray set:
DisplayMonitorIcon 0

To display the !M in the system tray set:
DisplayMonitorIcon 1

In both cases, then restart the server:

```
/etc/NX/nxserver --restart
```

By default NoMachine issues a balloon message to notify about events like user disconnections or user requests to connect. You can disable notification messages by setting the following key in the node configuration:
DisplayMonitorNotifications 0

TIP

If the displaying of monitor notification messages is disabled, the desktop owner will be unable to accept connection's requests by other users. Configure trusted users if you need to allow the connection without explicit authorization.

7.6. Hiding the Whiteboard and Chat Tools

If you want to disable the Whiteboard from the Monitor menu, edit the node configuration file to have:
EnableWhiteboard 0

Then restart the server:

```
/etc/NX/nxserver --restart
```

7.7. Handling of Discovering of this Server on LAN

By default NoMachine Workstation broadcasts information to let other NoMachine computers to discover it on the local network. You can disable this feature via:
the User Interface
In the Settings -> Server > Ports panel, uncheck 'Advertise this computer on the local network'.

or in **the server configuration file**
set the following key in the server configuration:

EnableNetworkBroadcast 0

Then restart the server to make changes effective:

```
/etc/NX/nxserver --restart
```

8. Notifications to Users

8.1. Whiteboard and Custom Notifications

NoMachine provides an instant messaging tool, named **whiteboard** which allows also drawing, the sharing of files with connected users and fast-track access to file transfer. To access it, connect to the user's desktop and from the Monitor (!M icon) in your system tray click on 'Show the whiteboard'. Note that if multiple users are connected at the same time to the same session, they will all see the same message.

As an alternative, it's possible to issue a dialog in the connected sessions to show a custom message by sending it from command line.

Sending a message to all running sessions:

```
/etc/NX/nxserver --broadcast "Your message goes here"

or sending a message only to the session specified by its session id:

/etc/NX/nxserver --message "Your message goes here"
```

8.2. Greeting Messages (for Virtual Desktops)

It is possible to welcome users when the virtual desktop has started by issuing a greeting message, either only the first time the user logs-in or every time the user connects to the Workstation. Update the node configuration file by writing the text of your message in any of the following keys:

NodeFirstLoginGreeting "Welcome to your first NX session"
NodeLoginGreeting "Welcome to your NoMachine session"

9. Supported Connection Protocols and Authentication Methods

NX Protocol
connections by default use the NX protocol which is its own protocol for secure communication over the network. Encryption in the NX protocol is implemented using OpenSSL TLS/SSL, based on ECDHE-RSA-AES128-GCM-SHA256 as the default cipher suite. ECDHE-RSA-AES128-GCM-SHA256 is an AES (Advanced Encryption Standard) block cipher with 128 bits key in GCM (Galois/Counter Mode). RC4 (ECDHE-RSA-RC4-SHA cipher suite) is used as a backward compatibility when connecting from or to versions 4.0.

When using the NX protocol, NX data can travel on TCP and UDP streams, even at the same time. The client and server can decide dynamically what transport to use, based on the type of data and the network conditions. Client and server negotiate the UDP transport at session startup, after having negotiated the main TCP link. UDP uses symmetric Blowfish encryption, with key negotiated on the secure TCP link. UDP is presently not available when using the SSH tunneling to ensure that all data goes through the same SSH link, as it was in legacy version 3. UDP protocol can be also disabled.

SSH Protocol
NoMachine Workstation also provides tunneling of connections using SSH and full integration with any authentication backend supported by the host SSH server.

Authentication methods
These are the authentication methods supported by NoMachine when connections use the NX protocol or the SSH protocol:

Authentication method	NX protocol	SSH protocol
Login with user's password	yes	yes
Login with SSH private key	yes	yes
Login with SSH private key provided by SSH agent (available since v. 6.3.6)	-	yes
Login with SSH private key private key stored on a PKCS11 smart card	-	yes
Login with Kerberos ticket	yes	yes
Support for SSH agent forwarding	-	yes
Support for Kerberos tickets authentication forwarding	yes	yes
Support for two-factor authentication	yes	yes

9.1. Defining Protocol in Server Configuration

Protocols are defined in the ClientConnectionMethods key in the server configuration. They are specified as a comma-separated list of values:

```
ClientConnectionMethods NX,SSH,HTTP
```

This key is automatically populated during the installation or the update of the package. It is possible to exclude any of the available protocols to force users to connect by the desired protocol.

For example, to use only NX protocol, set this key to:

```
ClientConnectionMethods NX
```

and restart the server to make changes effective:

```
/etc/NX/nxserver --restart
```

TIPS

I If your server supports SSH but it still reports that SSH is not available, check the ClientConnectionMethods key and ensure that the SSH values is set. Then restart the server.

II Removing 'HTTP' from the ClientConnectionMethods key will disable the starting of the NoMachine HTTP server and prevent connections via web.

9.2. Locking Down the Accepted Authentication Methods

Administrators may decide how the user should authenticate on the server by defining which authentication method(s) is/are available. Authentication methods can be set in the server configuration file by editing this key:

```
AcceptedAuthenticationMethods all
```

By default all methods are accepted. They can be restricted by providing a comma-separated list of values, they will indicate which authentication method is permitted.

Accepted values for **connections by NX protocol** are:
NX-password to allow password authentication.
NX-private-key to allow key-based authentication.
NX-kerberos to allow Kerberos ticket-based authentication.

while for **connections by SSH protocol**:
SSH-system to allow all methods supported for the system login. SSH authentication methods for the system login have to be set on the system for example in the PAM configuration.

For example, to accept key-based and Kerberos ticket-based authentication for the NX protocol:

```
AcceptedAuthenticationMethods NX-private-key, NX-kerberos
```

Settings in the client UI
When you connect by the client to a remote computer, you can select the authentication method for your connection while creating it or change it later (right mouse click on the connection icon -> Edit connection -> Configuration).

9.3. Changing Port for the NX Protocol

The default setting of NoMachine is to run connections via the NX protocol on port 4000. On server side, the Network Server, nxd, is listening on port 4000. It's mandatory that this port is open between client and server to allow connections by NX protocol.

If you change the port for nxd, connecting users will have to specify the new value in their connection settings in the client User Interface.

It's possible to modify the port for nxd
from the User Interface
in the Server status -> Server preferences -> Network services User Interface

or in the **server configuration file** by editing this key:
NXPort 4000

Restaring the nxd service is necessary to make this change effective:

```
/etc/NX/nxserver --restart nxd
```

When NX protocol is used, **UDP** communication for multimedia is enabled by default. UDP traffic uses a range of ports between 4011 and 4999. These ports must be open between client and server. If they are not available, traffic will fall back to TCP communication. You can change the port range, define a comma-separated list of ports or a single port by changing value set for the following key in the server configuration:
UDPPort 4011-4999

9.4. Changing Port for the SSH Protocol

The default port used for the SSH protocol is 22 on Linux. On such platform NoMachine relies on the SSH server installed on the system. If your SSHD is configured to listen on a port different from 22 you need to align the NoMachine server configuration accordingly. Connecting users will have to specify such value in their connection settings in the client User Interface.

If your SSH server is listening on a port different than 22, change the SSH port in the NoMachine configuration

in the Server status -> Server preferences -> Network services User Interface

or in the **server configuration file** by editing this key:
SSHDPort 22

9.5. Connecting to a Server Behind a Firewall (UPnP Port Mapping)

Automatic discover of the NoMachine Workstation host is possible only when the server and the user's machine are on the same LAN. When the user connects over the internet or from a different network, it's mandatory to know the public (or external) IP of the Workstation.

When the server is behind a firewall, you have to configure the router to forward external port to the nxd service (to use the NX connection protocol), to the SSH server (to use the SSH protocol) and to the nxhtd service (to connect by the web). By default the required ports are TCP ports: 4000 for NX, 4080 and 4443 for HTTP/HTTPS and UDP ports in the 4011-4999 range. Note that users will have to specify the external port in their connection settings in the client User Interface.

If the router on the server side supports UPnP/NAT-PMP, you can let NoMachine try to enable port forwarding in the router automatically. External ports will be selected randomly from the 20000 - 30000 range. Also in this case users will have to specify the external port in their connection settings in the client User Interface.

For connections by NX protocol, at session startup NoMachine will also try to map UDP ports by using UPnP.

Enabling the automatic port forwarding
Step 1: Set in the server configuration:
EnableFirewallConfiguration 1

Step 2: Specify for which service the port forwarding must be enabled by listing them in the following key:
EnableUPnP NX,SSH,HTTP

Step 3: Specify the port where the NX service will be redirected by editing respectively:
NXUPnPPort ""; SSHDUPnPPort "" and HTTPUPnPPort ""

TIP

To allow only connections by SSH (on external port 20048 for example) and use the automatic port forwarding, set in the server configuration:
ClientConnectionMethods SSH
EnableFirewallConfiguration 1
EnableUPnP SSH
SSHDUPnPPort "20048"
and restart the server.

You can enable port forwarding for connections by NX and HTTP/HTTPS protocol also **from the User Interface**
via the Settings -&glt; Server -&glt; Ports panel by selecting the service and enter its settings (click on 'Configure'). Then check the Gateway port option.

Retrieving information about UPnP port mapping
When the automatic port mapping is enabled, you can retrieve information about UPnP port mapping, e.g. IP of the gateway device, external port and port mapped by running:

```
/etc/NX/nxserver --upnpstatus
```

```
To terminate port mapping:

/etc/NX/nxserver --upnpunmap

To initiate port mapping:

/etc/NX/nxserver --upnpmap

You can also specify for how long port mapping should last by using

/etc/NX/nxserver --upnpmap --time
```

9.6. Using NoMachine DBs for Managing User Access

Use of NoMachine DBs can be configured by editing the server configuration. The table below reports which configuration key value has to be set to enable or disable specific behavior as defined in the 'Target' field:

Target	Server configuration key	Description
User's access based on system authentication (default)	EnablePasswordDB 0	Authentication is requested to the system, user's connection is allowed once the user has been authenticated. PAM, LDAP, AD are supported.
User access based on NX Password db	EnablePasswordDB 1	Authentication is verified against the NX password DB. Separate the NoMachine authentication from the system authentication. The user's account must exist on the system.
Allow connections from all authenticated users (default)	EnableUserDB 0	Every time a new account is created via NoMachine or an already existing system user runs the session for the first time, the user is added to the NoMachine NX Users DB, even when the use of NX Users DB is disabled. These users cannot be disabled and are always allowed to connect if they authenticate successfully.
Enable or disable user's access to NoMachine	EnableUserDB 1	By default all users are enabled to access the NoMachine system once authenticated. With this configuration a user can be disabled and re-enabled at any moment from command line.

10. User Management

10.1. Managing Users on the Workstation Host

To be able to login to the Workstation host, you need to have a valid account on that host, password cannot be empty. Your account can be a local account or a LDAP account or an AD account. NoMachine doesn't check if the source of users' account information is for example LDAP or local account database. Be sure to configure the login in advance.

You can create a local account by means of system tools, or via command line by using the 'nxserver' commands:

```
/etc/NX/nxserver --useradd USERNAME --system
```

Separate the system login from the NoMachine login
You can assign a password, specific for the NoMachine login only, different from system password to any valid account. To do that, enable NoMachine Password DB (EnablePasswordDB 1) in server.cfg. And ensure that the user is added to the NoMachine backend:

Â

```
sudo /etc/NX/nxserver --useradd USERNAME
```

Enabling and Disabling access for a NoMachine User
Prerequisites are:

i)The user has run at least one session **or** have been added to NoMachine dbs by means of '/etc/NX/nxserver --useradd' command.

ii) NoMachine Users DB is enabled (EnableUserDB 1 is set in server.cfg)

You can enable/disable a user and allow/forbid access to the Workstation for that user by running:

```
/etc/NX/nxserver --userenable USERNAME

or:

/etc/NX/nxserver --userdisable USERNAME

Note that 'nxserver --useradd USERNAME' adds the user to NoMachine dbs and automatically enables the user to log-in, while 'nxserver --userdel USERNAME' removes the user from NoMachine dbs and disables the user's ability to login by NoMachine.
```

Modifying the User's Password
You can modify the user's system password by running:

```
/etc/NX/nxserver --passwd USERNAME --system

or you can modify just the NoMachine password when Password db is in use(EnablePasswordDB 1 is set in server.cfg):

/etc/NX/nxserver --passwd USERNAME
```

Listing the NoMachine Users
All users who have run at least one session or have been added to NoMachine dbs are stored in the Users db. You can retrieve the complete list by running:

```
/etc/NX/nxserver --userlist
```

10.2. Connecting with a Privileged System Account

By default, NoMachine allows the running of sessions as privileged system user ('root' or 'sudo' on Linux). You can however configure the NoMachine Server to disallow it. Do it by disabling the following server configuration key:

EnableAdministratorLogin 0

To re-enable the possibility to log in as root or administrator, set:

EnableAdministratorLogin 1

10.3. Connecting as a NoMachine Trusted User

By default when the connecting user is different from the owner of the virtual desktop, the desktop owner has to authorize the user for the connection.

It is possible to define in advance a number of trusted users who don't need the specific owner's permission to connect to virtual desktops run by a different user.

In order to create a list of trusted users, administrators should use the nxserver commands for creating and editing users. These commands provide the --trusted option to define if the user is trusted for connections to the virtual desktop or not.

To create a new trusted user for virtual desktops on the system:

```
/etc/NX/nxserver --useradd USERNAME --system --trusted virtual

For example:

/etc/NX/nxserver --useradd nxtest01 --system --trusted virtual

To make an existing user trusted, modify trusted permissions or remove them:

/etc/NX/nxserver --useredit --trusted [virtual | physical | none]

For example edit user 'nxtest02' and give it trusted authorization for virtual desktops:

/etc/NX/nxserver --useredit nxtest02 --trusted virtual

To remove trusted authorization for user 'nxtest01':

/etc/NX/nxserver --useredit nxtest01 --trusted none
```

In a similar way, you can make a user trusted for virtual desktops and for connections to the physical desktop or for connections to the physical desktop only:

```
/etc/NX/nxserver --useradd USERNAME --system --trusted

/etc/NX/nxserver --useradd USERNAME --system --trusted physical
```

Make a user trusted for specific users' desktops You can assign the 'trusted' flag and make the user trusted only for those desktops owned by a specific user or list of users. For example, if a new user (userB) should be created on the system and made trusted only for desktops of userA:

```
/etc/NX/nxserver --useradd userB --system --trusted virtual --per-user userA

It's also possible to specify more users in a comma-separated list, e.g.:

/etc/NX/nxserver --useradd userB --system --trusted virtual --per-user userA,userC,userD
```

To list only trusted users:

```
/etc/NX/nxserver --userlist --trusted
```

11. Session Management

Each session on the same server is uniquely identified by a **session id** (which can look like:

B253864E822F5A235825F3AB8853AF00) and a **display id** (e.g.,1002).

A session on the NoMachine Workstation can be in any of the following statuses:
Connected - when it's connected to the remote display.
Disconnected - this status is available only for virtual desktop sessions and custom sessions. A session is marked as disconnected when it's disconnected from the remote display. A disconnected session can be reconnected at any time even from a different machine (migration). While a session is disconnected, applications on the remote server stay running. **Finished** - the session has been closed in a clean way and all NoMachine processes have been shut-down smoothly.
Failed - any of the NoMachine processes has failed to start or it has been "un-cleanly" terminated.
Transitional statuses are **Connecting**, **Disconnecting** and **Terminating**.

NoMachine Workstation is able to manage different types of sessions, named internally as in the table below. You can see the complete list of supported session types by running:

```
/etc/NX/nxserver --resourcelist --class session
```

Session types supported by the Workstation and their descriptions are:

Session type	Description
physical-desktop	Connect to the physical desktop of the Workstation host.
unix-xsession-default	Run the default virtual desktop as set on the system.
shadow	Connect to a virtual desktop session (desktop sharing/collaboration).
unix-console	Run a virtual Unix console application. It can be embedded into the client session window or be a floating window console depending on the user's choice: run or not the custom session in a virtual desktop
unix-desktop	Run a virtual custom application embedded into the player session window.
unix-application	Run a virtual custom application. It can be embedded into the client session window or be a floating window application depending on the user's choice: run or not the command in a virtual desktop.
unix-gnome	Run a virtual GNOME desktop. The ConnectPolicy key in the server configuration must have 'desktop=1' set.
unix-kde	Run a virtual KDE desktop. The ConnectPolicy key in the server configuration must have 'desktop=1' set.
unix-xdm	Run a virtual desktop through the X Desktop Manager.
unix-default	Run a virtual session by using the default X client script on server.
unix-script	Run a virtual session by using the X client script on server as specified by path.

11.1. Monitoring Sessions

You can monitor sessions from command line tools. Below are the server commands to be run from xterm or console.

Listing Running Sessions
To list all the running sessions, their display, session owner, remote IP of the connected client, session ID and session host:

```
/etc/NX/nxserver --list

You can also filter results on a per-user basis:

/etc/NX/nxserver --list USERNAME

or gather further information about connected clients:

/etc/NX/nxserver --list --client-version --client-platform
```

The number of active connections on the server corresponds to the number of sessions in status Connected. Session status is shown in the output of session history command.

Session History
History is preserved for a certain amount of time as set in the server configuration (SessionHistory key). To see the complete list of sessions, including those that have been cleanly terminated or failed, run:

```
/etc/NX/nxserver --history

To redirect the output of the session history to a file (available since v. 6.7):

/etc/NX/nxserver --history --file FILE

If you want to filter results on a per-user basis:

/etc/NX/nxserver --history USERNAME

or to get more details about a session:

/etc/NX/nxserver --history SESSIONID
```

Debugging a Failed Session with Session History
If a session is marked as failed in the session history output, the following command should provide information about the reason of the failure. Since v. 6.7 the output of the following commands has been extended to provide a short report helpful for a preliminary debug of the problem:

```
/etc/NX/nxserver --history SESSIONID

To redirect the error report to a file:

/etc/NX/nxserver --history SESSIONID --file FILE
```

Retrieving Statistics about Sessions with Session History
This feature elaborates a number of information about sessions, contained in the current session history. For example the number of sessions started, terminated, running and failed and their average startup time. The command to retrieve statistics is:

```
/etc/NX/nxserver --history --stats

To redirect statistics to a file:

/etc/NX/nxserver --history --stats --file FILE
```

Clearing Sessions History
You can reset the history backlog by running the following command.

```
/etc/NX/nxserver --history clear
```

Configuring the Session History Backlog
Data is preserved for 30 days. You can modify that in the server configuration file by uncommenting and setting a different value for the following key:
SessionHistory 2592000

This key accepts the following values:
< 0 Never delete data from NX session history.
0 Disable NX session history.
> 0 Keep data in session history for this amount of seconds.

11.2. Managing Sessions

Disconnecting a Virtual Desktop Session from Command Line
You can disconnect a session, if it's a virtual desktop one, by running:

```
/etc/NX/nxserver --disconnect SESSIONID

or:

/etc/NX/nxserver --disconnect DISPLAYID

You can also disconnect all virtual desktops belonging to a specific user:
```

```
/etc/NX/nxserver --disconnect USERNAME
```

TIP

Take SESSIONID or DISPLAYID from the output of the 'nxserver --list' command, they are the 'Session ID' and 'Display' column respectively. The same output also shows the user's name.

Disconnecting or Terminating Virtual Desktops Automatically

To disconnect a virtual desktop or custom sessions after a certain time of inactivity, uncomment and set a proper timeout value, in seconds, in the following node configuration key. For example, if you want to terminate sessions after 10 minutes of inactivity you need to set:
DisplayServerExtraOptions "-timeout 600"

If the NoMachine display agent doesn't receive any input from the user in the given timeout, it will either disconnect or terminate the session. Termination of the session will be carried out if the session is not persistent or no X application is connected to the display. Otherwise the agent will disconnect the session so that the X applications will be left running.

Note that the DisplayServerExtraOptions key is only for virtual desktops or custom sessions with X11 vector graphics enabled (default).

For web sessions, sessions connected to a virtual desktop (sharing of the virtual desktop), virtual desktops with X11 vector graphics disabled and connections to the physical desktop, set instead:
DisplayAgentExtraOptions "-timeout 600"

Terminating the Session from Command Line

To terminate a virtual desktop or custom session:

```
nxserver --terminate SESSIONID

or:

/etc/NX/nxserver --terminate DISPLAYID

You can also terminate all sessions belonging to a specific user:

/etc/NX/nxserver --terminate USERNAME

If you want to terminate all sessions, just restart the server:

/etc/NX/nxserver --restart

or if you want to terminate all sessions and forbid new connections until the server is started again:

/etc/NX/nxserver --shutdown

To start the server after a shutdown:
```

```
/etc/NX/nxserver --startup
```

Terminating Automatically Virtual Desktop/Custom Sessions in Status Disconnected
It's possible to specify for how long the server has to keep alive virtual desktops in the disconnected status. When the time has expired, the server will terminate virtual desktops if no user is connected there. To let the server terminate a disconnected virtual desktop after XXX seconds, edit the server configuration file, uncomment and set the timeout value (XXX) expressed in seconds in the following key:

```
DisconnectedSessionExpiry XXX
```

For example, by setting:

```
DisconnectedSessionExpiry 600
```

a virtual desktop will terminate after ten minutes provided there is no activity.

Terminating Automatically Virtual Desktop/Custom Sessions when the Maximum Number is Reached
To terminate a disconnected session when the maximum number of virtual desktops (see 'Limiting the Number of Virtual Desktops' below) is reached and make room for a new virtual desktop or custom session, enable the following key in the server.cfg file:

```
EnableAutokillSessions 1
```

Limiting the Number of Virtual Desktops or Custom sessions
You can set a limit for the number of virtual desktops provided that such limit does not exceed the number of connections allowed by the server license value (it's the 'Virtual Desktops' field in the server.lic file). NoMachine Workstation allows up to four concurrent virtual desktops.

For example to configure the server to allow only two concurrent virtual desktops, edit the server configuration and set:

```
VirtualDesktopsLimit 2
```

You can also specify how many virtual desktops a single user may run. For example, to allow 1 connection per-user, uncomment and set the following key in the server configuration file:

```
VirtualDesktopsUserLimit 1
```

A Practical Example
Limit the number of virtual desktops to three and keep alive a virtual desktop (inactive & disconnected) for one day. If a new virtual desktop is requested, the server will terminate the oldest virtual desktop in status disconnected to make room for the new session.

```
VirtualDesktopsLimit 3
EnableAutokillSessions 1
DisconnectedSessionExpiry 86400
```

Automatic Disconnection of Users
The automatic disconnection is a server configuration to rule the server behavior when the limit of users is exceeded and a new user is requesting to connect.

Current options are:
enabled (1): the server will automatically disconnect the user to make room for the new user.
disabled (0): the server will issue a pop-up message before disconnecting the user. The current user

can accept or not to disconnect itself. If no choice is made, the server will automatically disconnect this user and let the incoming user to connect.

The automatic disconnection applies when the maximum number of available connections to the desktops or the maximum number of available virtual desktops is exceeded.

To enable the automatic disconnection set the following key in the server.cfg file:
AutomaticDisconnection 1

To let the connected user decide or refuse to disconnect, set:
AutomaticDisconnection 0

Disabling persistent virtual desktops
Users may be forced to terminate their virtual desktop session by setting in the server configuration:
DisablePersistentSession all

In this way when the user closes the virtual desktop, the session is terminated instead of being disconnected. This server configuration key also accepts a list of comma-separated usernames and will be applied to the specified users. Non persistent sessions cannot be reconnected.

11.3. Setting one or more Virtual Desktop Environments

Pre-requisite to connect by NoMachine is that a desktop environment is installed on the system even if the host is headless or not started in graphics mode.

By default when the user chooses to create a new virtual desktop, NoMachine runs the default X session set on the system.

During its installation, NoMachine detects the default desktop environment set on the system and configures the node accordingly. Path and command to start the system desktop environment is defined in node.cfg by the DefaultDesktopCommand key. The Workstation is able to detect GNOME, Unity, KDE, LXDE and Xfce. If you have a different desktop environment, you can edit the DefaultDesktopCommand key accordingly.

For example to run MATE:
DefaultDesktopCommand "/usr/bin/mate-session"

or to run Pantheon:
DefaultDesktopCommand "/usr/bin/gnome-session --session=pantheon"

If there are multiple desktop environments installed, you can configure NoMachine to provide users the list of all the available desktops. To do that, set xsessions=1 in the ConnectPolicy key in the server configuration (server.cfg). By default this is disabled (xsession=0). To enable it, set:
ConnectPolicy
autocreate=1,autoconnect=1,automigrate=1,desktop=0,dialog=0,**xsessions=1**,udp=1

If there are GNOME and KDE installed, and/or if you want to provide XDM desktops, you can set desktop=1 in the ConnectPolicy key above.

To run GNOME and/or KDE, specify the command in the corresponding keys in node.cfg, for example

```
CommandStartGnome "/etc/X11/Xsession 'gnome-session --session=gnome'"
CommandStartKDE "/etc/X11/Xsession startkde"
```

11.4. Activating the Disconnect/Terminate Dialog

Terminating a virtual desktop session is done at the system level of the Workstation host. For example, if you are running a virtual desktop session (Gnome), you terminate the session by choosing the logout option from Gnome's system menu.

Disconnecting the session is done by clicking the 'X' button in the upper corner of the window. Alternatively, you can disconnect via the NoMachine session menu panel (Ctrl-Alt-0 -> Connection -> Disconnect).

To facilitate users, it's possible to display a dialog to let the user decide whether to disconnect or terminate the virtual desktop session when clicking on the X button which closes the session window.

The Disconnect/Terminate dialog is available for:

I virtual desktops running in X11 vector graphics mode
II virtual desktops **not** running in X11 vector graphics mode
III virtual custom sessions

To enable the Disconnect/Terminate dialog, enable the 'dialog' option (i.e. set dialog=1) in the following key in server.cfg:

ConnectPolicy autocreate=1,autoconnect=1,automigrate=1,desktop=0,**dialog=1**,xsessions=0,udp=1

11.5. Executing Custom Scripts on Server/Node Events

The **server configuration** provides a number of keys that can be activated to execute a custom script upon a certain event. According to the event, a number of parameters can be specified for each script. In a similar way, a number of keys is present in the **node configuration** file to allow to execute a custom script on a certain NoMachine node event. In both cases and according to the event, a number of parameters can be specified for each script.

Available for	Configuration key	Accepted parameter (server.cfg)	Accepted parameter (node.cfg)
server	UserScriptBeforeLogin	remote ip	-
server	UserScriptAfterLogin	username, remote ip	-
server	UserScriptAfterLogout	username, remote ip	-

server,node	UserScriptBeforeSessionStart	session id, username, node host, node port, main session id(*), main session type(*)	session id, username, session type, display, main session id(*), main session type(*)
server,node	UserScriptAfterSessionStart	session id, username, node host, node port, main session id(*), main session type(*)	session id, username, session type, display, main session id(*), main session type(*)
server,node	UserScriptBeforeSessionDisconnect	session id, username, node host, node port	session id, username, session type, display
server,node	UserScriptAfterSessionDisconnect	session id, username, node host, node port	session id, username, session type, display
server,node	UserScriptBeforeSessionClose	session id, username, node host, node port, main session id(*), main session type(*)	session id, username, session type, display, main session id(*), main session type(*)
server,node	UserScriptAfterSessionClose	session id, username, node host, node port, main session id(*), main session type(*)	session id, username, session type, display, main session id(*), main session type(*)
server,node	UserScriptBeforeSessionReconnect	session id, username, node host, node port	session id, username, session type, display
server,node	UserScriptAfterSessionReconnect	session id, username, node host, node port	ssession id, username, session type, display
server	UserScriptBeforeSessionFailure	session id, username, node host, node port,main session id(*), main session type(*)	-
server,node	UserScriptAfterSessionFailure	session id, username, node host, node port,main session id(*), main session type(*)	session id, username, session type, display, main session id(*), main session type(*)
server	UserScriptBeforeCreateUser	username	-
server	UserScriptAfterCreateUser	username	-
server	UserScriptBeforeDeleteUser	username	-
server	UserScriptAfterDeleteUser	username	-
server	UserScriptBeforeEnableUser	username	-
server	UserScriptAfterEnableUser	username	-
server	UserScriptBeforeDisableUser	username	-
server	UserScriptAfterDisableUser	username	-

(*) 'main session id' and 'main session type' parameters are available only when the user connects to an already running virtual desktop (session shadowing). They indicate respectively the id and type of the session to which the user is connected with his/her own session qualified by 'session id' and 'session type'.

A further key in the node configuration file, allows to run a custom script triggered on change resolution events (resize of the remote screen). The related key is:
UserScriptAfterRemoteResize

Note that the order of parameters is relevant. For example, a custom script to be run on node event 'UserScriptBeforeSessionStart' should use the $2 variable to retrieve username and $4 to retrieve display.

Pre-requisites to run custom scripts
Custom scripts must be executable. Custom scripts set-up in server.cfg are common to all the users who are accessing the server and are executed by the nxserver program. Since nxserver is running as the nx user, you have to grant this user the necessary permissions in order to execute the custom script.

Custom scripts set-up in node.cfg are executed by the nxnode program, which is run as the connected user. Place the script in a directory that is accessible by the node, i.e. accessible by the connected user(s).

By default if the execution of the scripts fails, the nxserver and nxnode will terminate. This means that the user's session will not start. You can override this behavior by forcing exit 0 inside the custom script and let the session start even if the custom script is failed.

TIP

If NoMachine Workstation is federated under a Cloud Server consider that custom scripts have to be placed in server.cfg or node.cfg file on the Workstation host, not on the Cloud Server.

12. Collaborative Virtual Desktops and Connections to the Physical Desktop

Multiple connections to a virtual desktop
By default users can connect to their virtual desktops and to virtual desktops owned by other users. When the desktop owner is different from the connecting user, he/she is always required to authorize the incoming request for connection. Authorization is not requested when the incoming user and the desktop owner are the same. This allows different users sharing the same instance of the virtual desktop and access all applications and resources interactively or in view-mode only. This feature is suitable for collaborative sessions and desktop sharing.

Request for desktop owner's authorization and interaction level can be configured
via the User Interface
You can configure how users will connect to a desktop owned by another user from the Settings -> Server -> Security panel. You can basically determine if users can connect or not without asking the desktop owner's permission and if users will be able to interact with the desktop. Allowing

connections in interactive mode grants the user full access to the desktop resources and applications. View-only mode is suggested for example when making presentations or teaching a lesson.

or **in the server configuration file**
besides using the graphical tools, you can configure the server by editing the server configuration file, uncommenting and setting a proper value for keys as illustrated in the following paragraphs.

TIPS

I Configurations made from the User Interface apply to connections to physical and virtual desktops. If you want to set a separate configuration for these desktops, you have to edit the server configuration manually.

II Rather than allow all users to connect without virtual desktop's owner authorization or click accept for every single user which would like to connect, it is possible to define in advance a number of trusted users who don't need the specific owner's permission.

III When the Workstation is federated under a Cloud Server, each user must have the same system account on the Workstation host and on the Cloud Server host. Password can be different.

Connect to the physical desktop

NoMachine Workstation supports the **screen blanking** feature: when active, the local user will see a black screen on the physical monitor while somebody is connected from remote to the physical desktop. Operations made on the physical screen are not shown and the local user cannot interact with the desktop until the remote user logs-out. Control is given back to the local user once the remote user has logged off. Screen blanking is available for physical hosts, it is not supported on virtual machines since it has effect on the physical monitor

You can activate the screen blanking feature on the Workstation host machine
via the User Interface:
in the Settings -> Server -> Security panel select the 'Blank the physical screen when somebody connects' option.

or **in the server configuration file**.
Uncomment and set:
EnableScreenBlanking 1

To disable the screen blanking, set:
EnableScreenBlanking 0

In both cases then restart the server to make this change effective:

```
/etc/NX/nxserver --restart
```

The screen blanking feature can be used in conjunction with the **automatic lock of the remote screen**. Even if the user didn't lock the screen before disconnecting by NoMachine, as soon as the screen is unblanked, the system lock screen will be activated automatically to keep the remote desktop protected even when the computer is running unattended.

You can enable the automatic remote screen lock **from the User Interface**
In Settings -> Server -> Security panel select the 'Lock the physical screen on disconnect' option

or **in the server configuration** file, server.cfg.
Uncomment and set:
EnableLockScreen 1

To disable the automatic screen lock, set:
EnableLockScreen 0

Then restart the server to make this change effective:

```
/etc/NX/nxserver --restart
```

12.1. Disabling Connections to Virtual Desktops

By default users can connect to virtual desktop sessions owned by a different user. To forbid this capability, set in the server configuration file:
VirtualDesktopSharing 0

TIP

This setting disables also the listing of other user's virtual session in the client User Interface.

12.2. Configuring Interaction Level to Virtual Desktops

To forbid users to interact with the desktop once connected set in the server configuration:
VirtualDesktopMode 0

In this way, the connected user will access the virtual desktop in view-only mode.

To allow interaction instead, ensure to have:
VirtualDesktopMode 2

It's also possible to allow users to interact with the desktop, except for resize operations. This corresponds to:
VirtualDesktopMode 1

12.3. Configuring Authorization to Connect to Virtual Desktops

By default, only system administrators, NoMachine administrators and NoMachine trusted users can connect to another user's virtual desktop without the explicit authorization of the desktop owner. This corresponds to the following setting in server.cfg:
VirtualDesktopAuthorization 1

To restrict access further, so that only NoMachine trusted users can connect without the desktop owner's authorization, set:
VirtualDesktopAuthorization 2

To allow users connecting to the virtual desktop without explicit permissions, set:
VirtualDesktopAuthorization 0

Settings above apply to **all users**.

TIP

To restrict the access without owner's authorization to given users, set: VirtualDesktopAuthorization 2 in the server configuration and assign the 'trusted for virtual' flag to the user:

```
/etc/NX/nxserver --useredit USERNAME --trusted virtual
```

12.4. Connections to Physical Desktop

By default, connections to the physical desktop are enabled but restricted to system administrators and NoMachine administrators and trusted users. They can connect without the explicit authorization of the desktop owner. Such configuration can be helpful for example to access the computer for remote administration when it runs unattended. This configuration corresponds to the following setting in server.cfg:
PhysicalDesktopSharing 2

To completely disable connections to the physical desktop, set:
PhysicalDesktopSharing 0

To enable connections to the physical desktop for all users, set:
PhysicalDesktopSharing 1

Enable or disable requesting the desktop owner's authorization
By default, the authorization of the desktop owner is requested to allow the user to connect, except for system administrator, NoMachine administrators and NoMachine trusted users. All the other users require the approval of the desktop owner. This corresponds to the following setting in server.cfg:
PhysicalDesktopAuthorization 1

To request the owner's authorization also for administrators and allow only NoMachine trusted users to connect without explicit authorization set:
PhysicalDesktopAuthorization 2

To never request the desktop owner's authorization and allow all users connecting to the physical desktop without explicit permissions, set:
PhysicalDesktopAuthorization 0

Limit user's interaction with the physical desktop
By default, users connected to the physical desktop can fully interact with it. This corresponds to the following setting in server.cfg (default):
PhysicalDesktopMode 2

To forbid users to interact with the desktop once connected (view-only mode) set in server.cfg:
PhysicalDesktopMode 0

A further possibility is to allow interaction except for resize operations:
PhysicalDesktopMode 1

System administrators
A privileged system user has to be defined by means of system tools.

The Workstation allows by default administrative users to connect. You can disable it by setting in the server configuration:
EnableAdministratorLogin 0

To re-enable the possibility to log in as root or administrator, set:
EnableAdministratorLogin 1

NoMachine administrators
Permissions as NoMachine administrator are totally indipendent from system, they don't alter the privileges of the user's account. To add the 'NoMachine administrator flag to an existent user, execute:

```
/etc/NX/nxserver --useredit USERNAME --administrator --yes
```

NoMachine trusted users
To assign the 'trusted' flag to an existent user, execute:

```
/etc/NX/nxserver --useredit USERNAME> --trusted physical
```

Override user's setting to disable 'Accept connection'
If the user disabled 'Accept connections' from the !M monitor menu of the Workstation, it's still possible to re-enable it via command line.

Execute from a terminal:

```
sudo /etc/NX/nxserver --useredit USERNAME --screensharing yes
```

13. Device Sharing, Copy&Paste and File Transfer

The Workstation allows users to access and share their devices and resources from local to remote and vice-versa. Disks, printers, USB devices and more can be connected inside the session to easily access them from both client and server side. At present device sharing is not available with web sessions and requires to connect by NoMachine client.

Two-ways copy and paste is fully supported. Web sessions implements the NoMachine virtual clipboard provides for copying text from/to the session running in the browser and the local computer.

Download/upload files from the session to the local computer and vice-versa is also fully supported in client and web sessions, as well as drag and drop of a file from remote to local and from local to remote.

By default device sharing, copy&paste and file transfer are always permitted. You can however completely disable any of these services or disable it only partially, for example to prevent users from sharing their local printer in the NoMachine session but permitting them to use the remote printer.

13.1. Connecting Devices

NoMachine implements a self-contained infrastructure for making available physical and logical devices over the network from local to remote or vice-versa.

The NoMachine infrastructure for device sharing ensures that all services work out of the box without the need for any additional change or configuration. It is possible to connect disks, printers, USB devices, network port and smartcards.

Connecting devices is supported only by NoMachine client (web sessions don't support that). Devices can be connected through the NoMachine menu within the session (ctrl+alt+0 to open it). Connected devices can be disconnected during the life of the session and reconnected later. If option 'Export this *deviceName* at session startup' is checked in the menu panel, this device is automatically reconnected at the next session start-up.

Disabling device sharing
You can disable selectively the possibility to share a device

from the User Interface
in the Server User Interface -> Devices panel

or **in the node configuration file**
by editing the corresponding keys. The manual configuration also allows the service to be limited to one-way, for example forbid to connect a local printer to the remote host. The next paragraphs deal with manual node configurations in detail.

The available devices are:

Devices	Configuration	Technical details
	Local and remote disks can be connected and disconnected during the life of the session and navigated by file browsing. A disk connected as 'Public' is available to all users accessing that desktop. A private disk	This service uses FUSE, installed on

Disks	is available only to the user who connected it. Administrators can configure paths on the server where public and private disks will be mounted as well as specifying which disks on the server can be made available to users.	the Linux system by default. The nxfs and nxfsserver programs are used to mount disks.
Printers	Local and remote printers can be connected at any time (bi-directional printing). A connected printer is listed among the available printers when printing a document or similar. A printer can be connected to be 'Public', i.e. available to all users connected to that desktop, or private, for a specific user. It can be also configured to be the default printer.	This services uses the CUPS infrastructure present on the Linux system. A printer can be exported to the server only if the connected user is in the lpadmin group.
USB devices	USB devices such as disks, pendrives, webcam etc... are forwarded through the network. For example, when a USB device is forwarded from local (where the player is running) to remote, it becomes available on the remote side only.	This service is based only on the NoMachine USB Server (nxusbd) and drivers (the nxusb.ko kernel module for Linux) and doesn't require external tools.
Network ports	Service ports (such as Samba, CUPS, FTP, SSH, telnet and others) can be made available from local to remote and vice-versa via a virtual network interface.	This service relies on a NoMachine tool plus a standard driver.
Smart Cards	A smartcard reader can be forwarded from client to server side and makes smartcard authentication available within the session. The server host must support authentication via smartcard.	Support for authentication with smart card has been set-up by relying on the Public Key Infrastructure (PKI) and requires an OpenSC compatible smart card. It can be integrated with Kerberos ticket authentication and ticket forwarding.

13.2. Disks

NoMachine allows access to local and remote file systems from within the session through the SSHFS file-sharing protocol and by means of FUSE, a technology to implement a fully functional filesystem in a userspace program.

Connected folders and disks can be disconnected during the life of the session or left as they are.

By default, all disks from the server are available to be connected to the end-user's machine. However you can specify a set of disks and folders by editing a proper value for the DiskSharingList key in the node configuration file. The default value is: all. Alternatively, you can specify a list of comma-separated directories. Note that $(HOME) and $(USER) are accepted values.

Disks from the end-user's machine can be connected on the server in 'Public' or 'Private' mode.

Connecting public disks
Disks from the end-user's machine can be connected on the server in 'Public' or 'Private' mode.

By default public disks are exported from player to "$(PUBLIC)" directory on the server, where $(PUBLIC) is: **/media** on Linux.

You can specify a different path by un-commenting and editing the DiskSharingPublicBasePath key in the node configuration file.
Note that $(USER) is an accepted value that can be also concatenated to specify the path to a directory, for example "/tmp/$(USER)".

The target directory must exist on the system!

Disabling Disks' Connection
To forbid disk and filesystem sharing, uncomment and set a proper value for the EnableDiskSharing key in the node configuration file:
client The filesystem on the client can be connected to server side and accessed from the session.
server The filesystem on the server can be connected to the end-user's machine and accessed through the whole life of the session.
both Client and server filesystem can be connected to remote and local sides respectively.
none Neither client or server filesystem can be connected.

For example, to forbid connecting disks from remote to local side, set in the node configuration:
EnableDiskSharing client

13.3. Printers

The printers sharing infrastructure integrates client-side printers with the server-side printing subsystem and vice-versa. Printers available on the client machine can be shared and used within the session as well as printers on the server side which can be made available on the end-user's machine.

Connected printers can be disconnected during the life of the session or left as they are. In this case, they are automatically shared at the next session start-up.

On Linux this service uses the CUPS infrastructure present on the system. With CUPS 1.4 or later, to ensure that users are able to connect a printer from local to their NoMachine session on Linux , it's necessary that the user already belongs to the CUPS System Group on the NoMachine server host. This is because in order to add a printer to the CUPS system, the 'lpadmin' command line tool has to be executed by a user who belongs to the CUPS's System Group, which can be for example 'lpadmin' on Ubuntu, 'sys' on Fedora, RHEL and CentOS distributions.

Disabling Printers' Connection
To forbid printer sharing it is necessary to uncomment and set a proper value for the EnablePrinterSharing key in the node configuration file:
client Printers on the client can be connected to server side and made available within the session.
server Printers on the server can be connected to the end-user's machine.
both Client and server printers can be connected to remote and local sides respectively.
none Neither client or server printers can be connected.

For example, to forbid a server-side printer to be connected to the end-user machine, set in the node configuration:
EnablePrinterSharing client

13.4. USB Devices

This service creates a USB tunnel between client and server to forward devices over the network such as hard disk, web cams, barcode readers, and pen drives from local to remote desktops and vice-versa.

Disabling USB Forwarding
To forbid USB device sharing it is necessary to uncomment and set a proper value for the EnableUSBSharing key in the node configuration file:
client USB devices on the client can be forwarded to server side and made available within the session.
server USB devices on the server can be connected to the end-user's machine.
both Client and server USB devices can be connected to remote and local sides respectively.
none Neither client or server USB devices can be connected.

For example, to avoid that users can forward a USB devices from the server to its own machine, set in the node configuration:

```
EnableUSBSharing client
```

13.5. Network Ports

NoMachine can create virtual network interfaces and establish a bridge between local and remote sides or vice-versa to provide transparent access to network resources.

This service allows access to any of the default network servers like Samba, CUPS, FTP, SSH and Telnet or any other type, for example a MySQL server.

Connecting a Samba server allows access to resources on that server host via the SMB/CIFS protocol. Connecting a local CUPS server to the remote side allows mounting of printers (local to the user) on that remote CUPS subsystem so that files can be printed on the remote side via the IPP protocol.

Some typical examples of usage:
Print to remote printers from the session
If you have a Linux or Mac machine you can add the local CUPS server via the player toolbar. Choose to add a local server and select CUPS. In this way all printers that are available on your side will be available also on the server and you can print all your documents via the native CUPS (IPP) protocol.

Access a remote host not in your Network Neighborhood
If the remote host has a Samba server, you can add it via the player toolbar. Choose to add a remote server and select Samba as server type. Once that Samba server is added, the remote host shows up in your local Network Neighborhood. You can then connect to remote folders via SMB/CIFS protocol as if that host was in your local network.

Make available a client side HTTP server
You can add your local HTTP server via the player toolbar and make it available on the remote host where your session is running. In this way you can develop and test your web application directly inside the session, without the need for sharing or moving files from remote to local.

Connect to MySQL server behind a firewall
You can choose to add a remote server via the player toolbar. Select 'Custom' and specify MySQL and the port for the MySQL server, by default 3306. Once done, you can connect to that MySQL server via the MySQL client installed on your PC.

Disabling Network Port Forwarding
To forbid network server sharing you must uncomment and set a proper value for the EnableNetworkSharing key in the node configuration file: **client** Network servers on client side can be connected and made available within the session.
server Network server on the server side can be connected and made available on the end-user's machine.
both Network servers from client and server side can be connected to remote and local sides respectively.
none Neither client or server side network servers can be connected.

For example, to forbid users from connecting their local ports to the server, set in the node configuration:
EnableNetworkSharing server

13.6. Smartcard Readers

When the smartcard reader plugged into the enduser's host is forwarded to the server host, the smartcard authentication is made available inside the session. It can be integrated on with Kerberos Ticket system for example for implementing single sign-on (SSO).

Disabling Smartcard readers' Forwarding
You can enable or disable support for smarcard forwarding by uncommenting and setting the EnableSmartcardSharing key in the node configuration to 1 or 0 respectively.

To disable it set in node configuration file:
EnableSmartcardSharing 0

13.7. Copy and Paste Operations

By default users can copy and paste from local to the session and vice-versa.

You can configure the server to limit such operations by setting proper values in the configuration file as explained below.

Limiting copy & paste operations
To forbid copy & paste partially or totally, uncomment and set a proper value for the EnableClipboard key in the server configuration file:
client Content copied on the user's side can be pasted inside the session.
server Content copied inside the session can be pasted on the user's side.
none No copy and paste operations are allowed.
both Two-way copy and paste operations are allowed.

Limiting the Clipboard Buffer
By default, the clipboard buffer is unlimited. If you want, for example, to limit the clipboard buffer to 4MB, you have to uncomment and set the following key (value is espressed in bytes) in the node configuration file:
ClipboardBufferLimit 4194304

13.8. Transferring Files

When a user is connected to the desktop, they have the possibility to transfer files by using the Connection Monitor tool from the system tray within the session. The user can transfer a file from their own PC to the remote host where the session is running and vice-versa. If multiple users are connected, each of them can send a file to a specific user or to all connected users. Drag and drop of a file is also supported

You can manage file transfer
from the User Interface
In the Server User Interface -> Transfers panel

or **via node configuration**.

Disabling File Transfer
To forbid file transfer you have to uncomment and set a proper value for the EnableFileTransfer key in the node configuration file:
client Files can be transferred from client machine to the server.
server Files can be sent from the server to clients.
both Client and server files can be transferred on remote and local respectively.
none Neither client or server files can be transferred.

For example, to forbid users from transferring a file from the server to their PC:
EnableFileTransfer client

14. Multimedia and Session Recording

14.1. Supporting Audio and Microphone

On Linux, NoMachine audio framework is integrated with PulseAudio sound server. If PulseAudio is not available on the system, NoMachine is able to use ALSA (Advanced Linux Sound Architecture). This is automatically managed by the NoMachine server so that multimedia support can work out of the box without the need for any configuration. If both PulseAudio and Alsa are available, the administrator might want to configure the node to use one or the other.

Disabling or Setting Audio Support
To disable audio and microphone support, uncomment and set the AudioInterface key to 'disabled' in the node configuration file:
AudioInterface disabled

On Linux it is possible to define whether PulseAudio Server or ALSA has to be used by setting AudioInterface key to 'pulseaudio' or 'alsa' respectively. For example:
AudioInterface pulseaudio

14.2. Recording your Screen

NoMachine can video-record all activity made inside the session or on the desktop. To start the recording of the session, users should open the NoMachine menu inside the session (ctrl+alt+0) and click on the 'Recording' button icon to access the Recording panel. From this panel it's possible to open the recording bar, change audio and video quality and open the recording directory to access all recorded files. Session recording is not available with sessions on the web.

To record activity made on the desktop, start the recording from the !M icon menu in the system tray of the Workstation host and show the Recording bar from there. Desktop activities can be registered on the physical desktop without the need to be connected by NoMachine.

Recorded files are saved by default in WebM format and can be played back directly with NoMachine or any other player supporting that format. Video streams can be encoded only with VP8 or H.264 when supported. Recorded files are saved by default on the user's device in the NoMachine directory under the 'Documents' directory.

Disabling session recording
To prevent users from recording their session activities, edit the node configuration to set:
EnableSessionRecording 0

Disabling desktop recording
To prevent users from recording desktop activities, even when physically logged into the Workstation host, edit the node configuration to set:
EnableLocalRecording 0

14.3. Automatic Screen Recording

The automatic recording of the session at session startup is disabled by default.

To enable it, execute in a terminal:

```
sudo /etc/NX/nxserver --recording yes
```

The automatic screen recording applies to both sessions connections to the physical display and virtual desktops.

TIP

You can also define the percentage of session to be recorded by specifying the '--percentage PERCENTAGE' parameter or activate the recording for a specific user only or group of users by means respectively of: '--user ' and '--group '.

15. Automatic Updates

The Workstation, as well as the other NoMachine client and server products, periodically checks NoMachine repositories (by default every two days) to verify if updates are available and will prompt a dialog informing the user that a new version is available.

It will **never** automatically update the current installation. Also the download in background of a new software version will not lead to an automatic update of the current installation.

A separate guide which deals specifically with all the possible options for the automatic software updates is available on the web site in this section:

https://www.nomachine.com/all-documents

16. Logging Facilities

To retrieve logs by using the NoMachine tools, please refer to guides available in the Configuration section at: https://www.nomachine.com/all-documents.

TIP

When debug mode is enabled, server logs may increase consistently. It's suggested to keep debug level only for the time necessary to reproduce the problem and collect logs.

17. Setting-up a Centralized Access to Multiple Workstation Servers

If you own multiple installations of Workstation, you may need to provide a single point of access to all of these servers. This can be done by installing NoMachine Cloud Server on a dedicated host and add each Workstation to it.

In this way, users will connect to the hostname/IP of the Cloud Server and will be redirected to the appropriate Workstation or, depending on the Cloud Server configuration, will be able to choose it manually.

You may also configure the NoMachine centralized infrastructure to make each Workstation accept or refuse direct connections to its host.

To grant high available access to this centralized system, it's possible to add a second Cloud Server to the first one and set-up a failover cluster.

TIP

NoMachine Cloud Server doesn't provide load-balancing of sessions among the federated servers. If you need to load-balance sessions, please consider a different solution, i.e. the Enterprise Terminal Server with Terminal Server Nodes.

17.1. Federating the Workstation Under a Cloud Server

In order to federate an Workstation under a Cloud Server, connect to the Cloud Server host as a NoMachine administrator and use the graphical interface to add the server.

Otherwise, execute on the Cloud Server host the 'nxserver --serveradd ' command.

Refer to the following article for more details about the Cloud Server multi-server environment, included instructions to set-it up and FAQs: https://www.nomachine.com/AR05R01090.