NetXMS User Guide

Release 5.2.0

Raden Solutions, SIA

Apr 28, 2025

CONTENTS

INTRODUCTION

1.1 About this document

The User Manual describes the main aspects of NetXMS monitoring system. This manual enables all users to get an overview of the various functionalities of NetXMS. The main aspects outlined in here describe the possibilities and functionaries of the NetXMS interface and elucidate working processes.

1.2 Target audience

This manual is intended for NetXMS operators, and provides all information necessary to successfully operate NetXMS.

1.3 Conventions

The following typographical conventions are used in this manual.

Sample	Description
Button	Any GUI element: Button, Menu item
Another Guide	Reference to external manual or man page
`Control-M`	Keyboard shortcut
DCI	Term which could be found in glossary
File ▸ Exit	Menu selection path, you must click on *File*, then *Exit*

BASIC CONCEPTS

2.1 Objects

All monitored network infrastructure is represented as a set of *objects* in NetXMS monitoring system. Each object represents one physical or logical entity (e.g. host or network interface), or group of them (e.g. subnet, container). Objects are organized into hierarchical structure. An object can have several parents, e.g. a node can belong to multiple containers, subnets and templates. Structure can be modified either manually or automatically with the help of Auto-bind scripts.

Each object has it's own access rights. Access rights are applied hierarchically on all children of object. For example if *Read* access right is granted to a user on a *Container*, then user has *Read* right on all objects that this *Container* contains.

Every object has set of attributes; some of them exist for all objects (like *id* and *name* or *status*), while other depend on object class - for example, only *Node* objects have attribute *SNMP community string*. In addition to the above mentioned attributes, it's possible to define custom attributes. This can be done by user in the Management Client, from NXSL script or by external application via NetXMS API.

NetXMS has seven top level objects - `Entire Network`, `Service Root` (named "Infrastructure Services" after system installation), `Template Root`, `Asset Root`, `Network Map Root`, `Dashboard Root` and `Business Service Root`. These objects serve as an abstract root for an appropriate object tree. All top level objects have only one editable attribute - name.

Object Class	Description	Valid Child Objects
Entire Network	Abstract object representing root of IP topology tree. All zone are located under it. System can have only one object of this class.	• Zone
Zone	Object representing group of (usually interconnected) IP networks without overlapping addresses. Contains appropriate subnet objects.	• Subnet
Subnet	Object representing IP subnet. Typically objects of this class are created automatically by the system to reflect system's knowledge of IP topology. The system places Node objects inside an appropriate Subnet object based on an interface configuration. Subnet objects have only one editable attribute - *Name*.	• Node

Table 1 – continued from previous page

Object Class	Description	Valid Child Objects
Service Root	Abstract object representing root of your infrastructure service tree. System can have only one object of this class. After system installation it is named "Infrastructure Services".	• Circuit • Chassis • Cluster • Condition • Collector • Container • Mobile Device • Node • Rack • Sensor • Wireless Domain
Collector	Object similar to container, but with data collection capabilities.	• Circuit • Cluster • Chassis • Condition • Collector • Container • Mobile Device • Node • Rack • Sensor • Wireless Domain
Container	Grouping object which can contain any type of objects that Service Root can contain. With help of container objects you can build object's tree which represents logical hierarchy of IT services in your organization.	• Circuit • Cluster • Chassis • Condition • Collector • Container • Mobile Device • Node • Rack • Sensor • Wireless Domain
Cluster	Pseudo-object defining any process: technological or logical that aggregates information from several separate nodes.	• Node
Circuit	Reference of multiple interfaces will allow to use this object to represent different types of network services beyond - multilink interfaces, links between sites, virtual circuits, etc.	• Interface
Rack	Object representing a rack. It has the same purpose as container, but allows to configure visual representation of equipment installed in a rack.	• Node • Chassis
Chassis	Object representing a chassis, e.g. a blade server enclosure. Chassis can be configured as a part of a rack.	• Node

Table 1 – continued from previous page

Object Class	Description	Valid Child Objects
Condition	Object representing complicated condition - like "cpu on node1 is overloaded and node2 is down for more than 10 minutes". Conditions may represent more complicated status checks because each condition can have a script attached. Interval for evaluation of condition status is configured in Server Configuration Variables as ConditionPollingInterval with default value 60 seconds.	
Node	Object representing physical host or network device (such as a router or network switch). These objects can be created either manually by administrator or automatically during network discovery process. They have a lot of attributes controlling all aspects of interaction between NetXMS server and managed node. For example, the attributes specify what data must be collected, how node status must be checked, which protocol versions to use, etc. Node objects contain one or more interface objects. The system creates interface objects automatically during configuration polls.	• Interface • Network Service • VPN Connector
Interface	Interface objects represent network interfaces of managed computers and devices. These objects created automatically by the system during configuration polls or can be created manually by user.	
Network Service	Object representing network service running on a node (like http or ssh), which is accessible online (via TCP IP). Network Service objects are always created manually. Currently, the system works with the following protocols - HTTP, POP3, SMTP, Telnet, SSH and Custom protocol type.	
VPN Connector	Object representing VPN tunnel endpoint, is used for interfaceless tunnels (like ipsec). Such objects can be created to add VPN tunnels to network topology known to NetXMS server. VPN Connector objects are created manually. In case if there is a VPN connection linking two different networks open between two firewalls that are added to the system as objects, a user can create a VPN Connector object on each of the firewall objects and link one to another. The network topology will now show that those two networks are connected and the system will take this condition into account during problem analysis and event correlation.	
Sensor	Logical object with data collection capabilities. NetXMS does not perform direct network communication with sensor, but data is collected by some other means, e.g. using MQTT protocol.	
Wireless Domain	Object representing wireless network, made up from one or several wireless controllers (represented by nodes with Wireless Controller capability) and thin access points.	• Access point • Node
Access point	Object representing thin wireless access point managed by a central controller. These objects are created automatically by the system.	
Template Root	Abstract object representing root of your template tree.	• Template • Template Group

Table 1 – continued from previous page

Object Class	Description	Valid Child Objects
Template Group	Grouping object which can contain templates or other template groups.	• Template • Template Group
Template	Data collection and agent policy template. See admin guide for more information about templates. If an object is a child of a template, this means that teplate is applied to that object.	• Acces point • Collector • Cluster • Mobile Device • Node • Sensor
Asset Root	Abstract object representing root of hardware asset management tree.	• Asset • Asset group
Asset Group	Grouping object which can contain assets or other asset group.	• Asset • Asset group
Asset	Hardware management asset	
Network Map Root	Abstract object representing root of your network map tree.	• Network Map • Network Map Group
Network Map Group	Grouping object which can contain network maps or other network map groups groups.	• Network Map • Network Map Group
Network Map	Network map.	
Dashboard Root	Abstract object representing root of your dashboard tree.	• Dashboard • Dashboard Group
Dashboard Group	Grouping object which can contain dashboards or other dashboard group	• Dashboard • Dashboard Group
Dashboard	Dashboard. Can contain other dashboards.	• Dashboard
Business Service Root	Abstract object representing root of your business service tree. System can have only one object of this class.	• Business Service • Business Service Prototype
Business Service	Object representing single business service. Can contain other business services or business service prototypes.	• Business Service • Business Service Prototype
Business Service Prototype	Prototype from which business service objects are automatically populated.	

2.1.1 Object status

Each object has a status. Status of an object calculated based on:

- Polling results
- Status of child objects (e.g. interfaces of node, nodes under container)

- Active alarms, associated with the object (after an alarm is resolved or terminated, it no longer affects object status)

- Value of status *DCIs* (DCI that has `Use this DCI for node status calculation` property enabled)

There are multiple options for status calculation, see admin guide for more information.

For some object classes, like Report or Template, status is irrelevant. Status for such objects is always *Normal*. Object's status can be one of the following:

Nr.	Status	Description
0	🟢 Normal	Object is in normal state.
1	⚠ Warning	Warning(s) exist for the object.
2	⚠ Minor	Minor problem(s) exist for the object.
3	🔺 Major	Major problem(s) exist for the object.
4	❌ Critical	Critical problem(s) exist for the object.
5	❓ Unknown	Object's status is unknown to the management server.
6	⊗ Unmanaged	Object is set to "unmanaged" state.
7	⊗ Disabled	Object is administratively disabled (only applicable to interface objects).
8	♻ Testing	Object is in testing state (only applicable to interface objects).

2.1.2 Unmanaged status

Objects can be unmanaged. In this status object is not polled, DCIs are not collected, no data is updated about object. This status can be used to store data about an object that is temporary or permanently unavailable or not managed.

2.1.3 Maintenance mode

This is special status, that's why it is not included in above status list. This status prevents event processing for specific node. While this node in maintenance mode is still polled and DCI data is still collected, but no event is generated.

2.2 Data Collection Items

From each node NetXMS can collect one or more *metrics* which can be either single-value (e.g. "CPU.Usage"), list (e.g. "FileSystem.MountPoints") or table (e.g. "FileSystem.Volumes"). When new data sample is collected, it's value is checked against configured thresholds. This documentation use term *Data Collection Item* (DCI) to describe configuration of metric collection schedule, retention, and thresholds.

Metrics can be collected from multiple data sources:

Source	Description
Internal	Data generated inside NetXMS server process (server statistics, etc.)
NetXMS Agent	Data is collected from NetXMS agent, which should be installed on target node. Server collect data from agent based on schedule.
SNMP	SNMP transport will be used. Server collect data based on schedule.
Web service	Data is objained from JSON, XML, or plain text retrieved via HTTP
Push	Values are pushed by external system (using *nxpush* or API) or from NXSL script.
Windows Performance counters	Data is collected via NetXMS agent running on Windows machine.
Script	Value is generated by NXSL script. Script should be stored in *Script Library*.
SSH	Data is obtained from output of ssh command executed through SSH connection.
MQTT	Data is obtained by subcribing to MQTT broker topics.
Network Device Driver	Some SNMP drivers (NET-SNMP, RITTAL as of NetXMS v. 3.8) provide parameters for data collection. E.g. NET-SNMP provides information about storage this way.

2.2.1 Thresholds

Each threshold is a combination of a condition and event pair. If a condition becomes true, associated "activation" event is generated, and when it becomes false again, "deactivation" event generated. Thresholds let you take a proactive approach to network management. Thresholds can be defined for any data collection items that is monitored, more than one threshold for a single DCI can be defined.

2.3 Events and Alarms

Many services within NetXMS gather information and generate events that are forwarded to NetXMS Event Queue. Events can also be emitted from agents on managed nodes, or from management applications residing on the management station or on specific network nodes. All events are processed by NetXMS Event Processor one-by-one, according to the processing rules defined in Event Processing Policy. As a result of event processing, some actions can be taken, and event can be shown up as alarm, sent as e-mail and notifications (SMS, instant messages). NetXMS provides one centralized location - the Alarm Browser, where the alarms are visible to your team. You can control which events should be considered important enough to show up as alarms. You and your team can easily monitor the posted alarms and take appropriate actions to preserve the health of your network.

Examples of alarms include:

- A router exceeded its threshold of traffic volume that you configured in Data Collection.

- The shell script that you wrote gathered the specific information you needed and posted it to the NetXMS as an event.

- One of your mission-critical servers switched to UPS battery power.

- An SNMP agent on a managed critical server forwarded a trap to NetXMS because it was overheating and about to fail.

2.4 Zones

As NetXMS server keeps track of an IP topology, it is important to maintain the configuration in which IP addresses do not overlap and that two IP addresses from same subnet are really within one subnet. Sometimes, however, it is needed to monitor multiple sites with overlapping IP address ranges. To correctly handle such situation, zoning must be used. Zone in NetXMS is a group of IP subnets which form non-overlapping IP address space. There is always zone 0 which contains subnets directly reachable by management server. For all other zones server assumes that subnets within that zones are not reachable directly, and proxy must be used.

USER INTERFACE

> **Note**
>
> One of the goals of NetXMS Management Client is to provide identical user experience across all supported platforms, including Web Interface. Screenshots in this particular guide are based on Windows version.
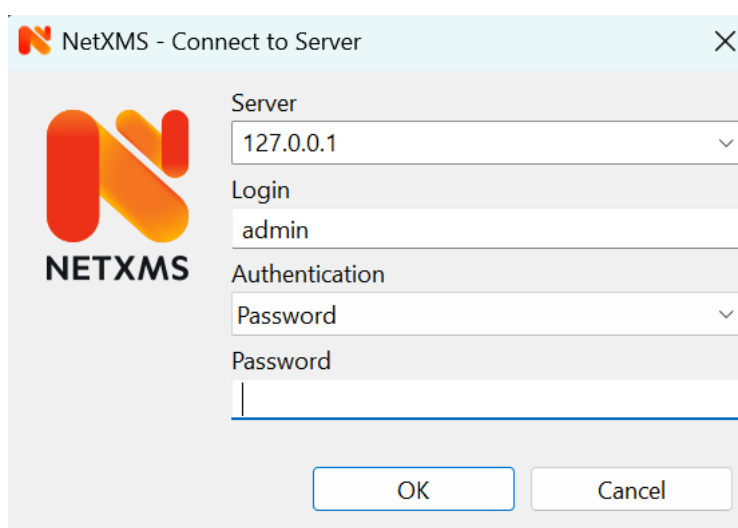
3.1 Login

Fig. 1: Login Dialog

When Management Client is started, user is presented with login dialog. User should enter server host name or IP address, login and password.
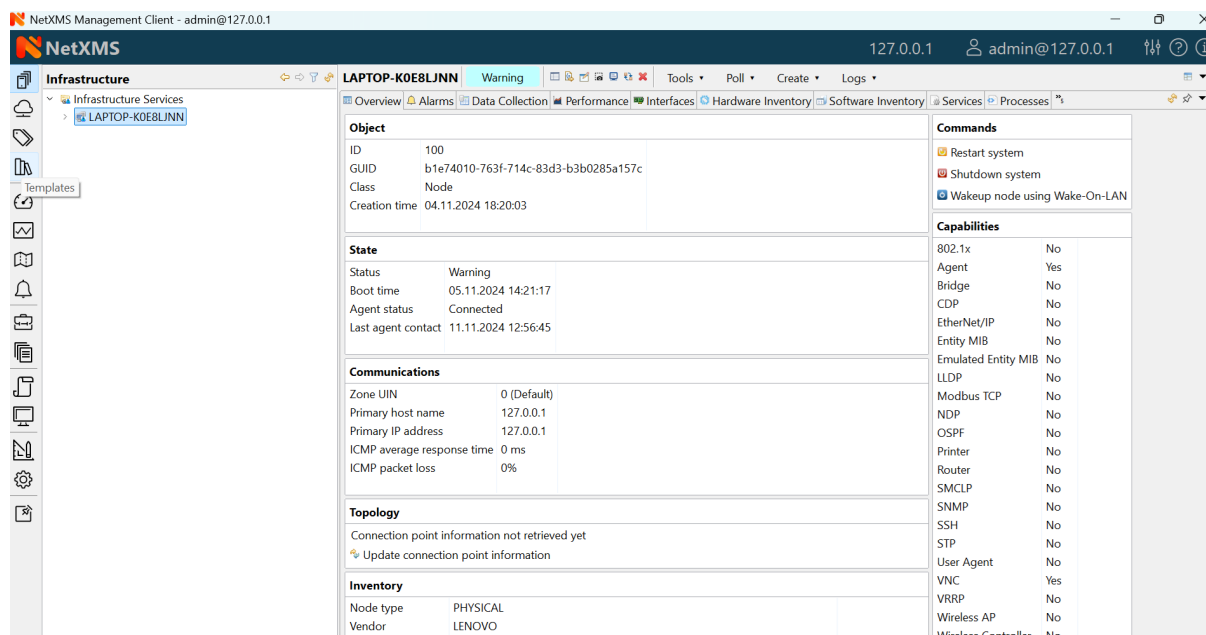
3.2 Perspectives

On the left hand side of Management Client there are perspective selection buttons. Each perspective serves its logical purpose, e.g. showing monitored objects, editing data collection templates, etc.

A perspective contains one or multiple *views*. If there are multiple views, they are organized as *View Stack* with tab navigation.

Some perspectives have a button in upper right corner which opens a view with additional object information.

Some views may have a toolbar above it on the right hand side - icons on the toolbar provide access to frequently needed actions. If available, a view's menu is displayed under vertical ellipsis icon on the toolbar.

Available perspectives are:

- Infrastructure (allows organizing objects in logical way, based on location, type of node etc)

- Network (organizes objects by zone and network subnet)

- Assets (hierarchical structure of available hardware assets)

- Templates (special objects with data collection and policy configuration)

- Dashboards (combines any available visualization components with data from multiple sources in order to create high-level views to see network (or parts of it) health at a glance.)

- Graphs (saved graphs with can display collected data from multiple objects)

- Maps (layer 2 topology, IP Topology or Custom maps)

- Alarms (alarm view for all objects)

- Business Services (tool for availability monitoring of logical services. Company email, web site, server farm, call center - all are examples of logical services.)

- Reporting (handles execution and rendering of reports by communication to separate reporting process)

- Logs (collective view of different logs)

- Monitor (event, trap and log collective display in real time)

- Tools (selection of powerful search and NetXMS server management tools)

- Configuration (provides access to configuration of the system)

- Pinboard (allows quick access to favorite views, for adding a view to pinboard see example below)

3.3 Object Browser

Some perspectives, e.g. Infrastructure, Network, Templates, etc… use object browser. Object browser represents objects as a tree. Tree is built based on object hierarchy and user permissions. Only objects available to currently logged in user will be shown. User has two options to interact with objects:

- Click Left mouse button to select object. Views on the right hand side provide information about currently selected object (see *Object details*).
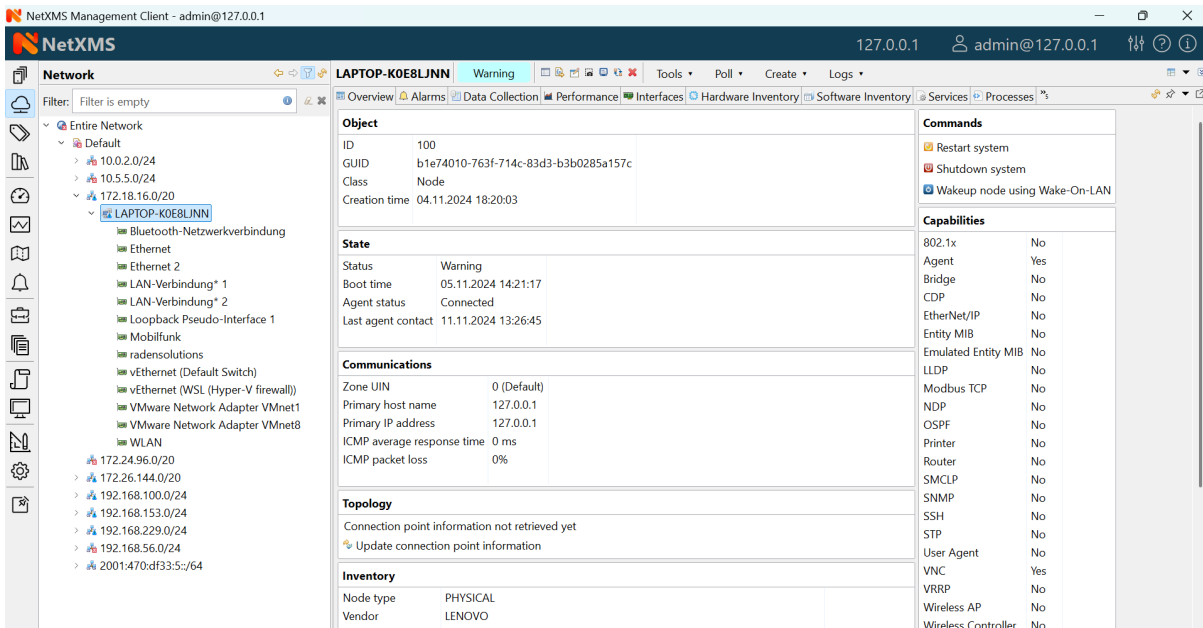
- Click Right mouse button to open context menu with actions available for this particular object type
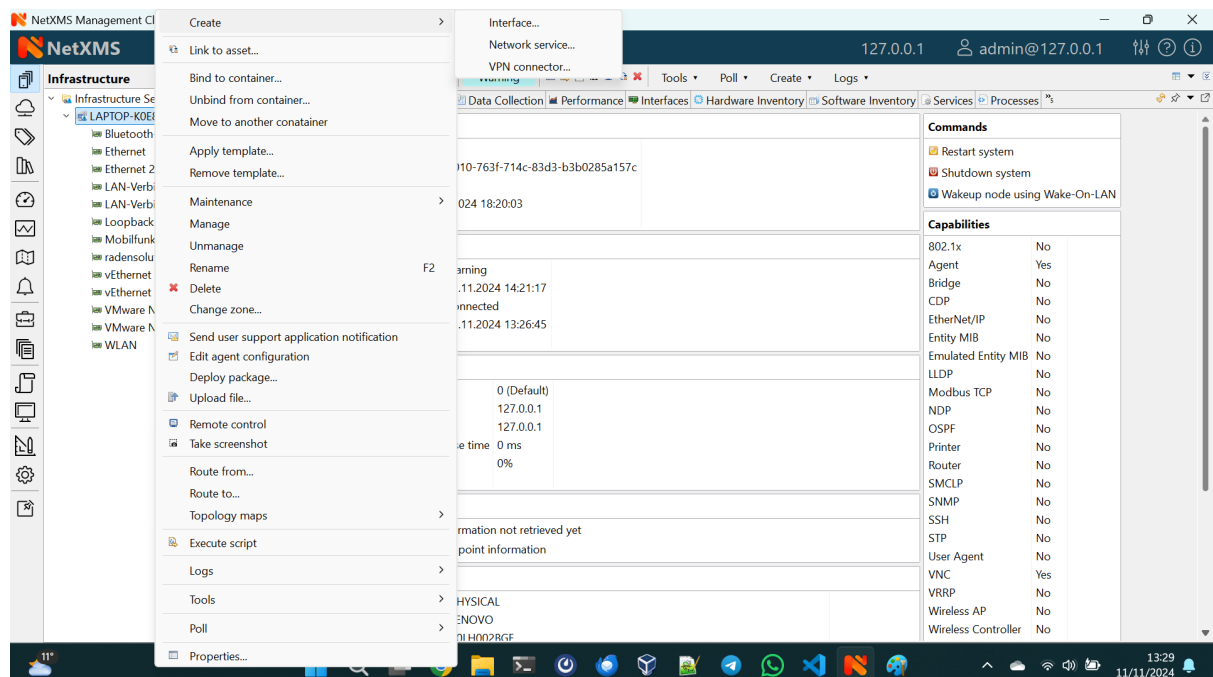
3.3.1 Object status

System track status of each object, which can range from *Minor* to *Critical*. Status is displayed as overlay on icon of each object.

3.3.2 Filtering

Above object tree there is filter field that allows to filter objects in the object tree. Filter supports a number of prefix characters that define how search is performed:

Prefix	Status
>	Search by IP address part
^	Search by exact IP address
#	Search by object ID
/	Search by comment
@	Search by zone ID

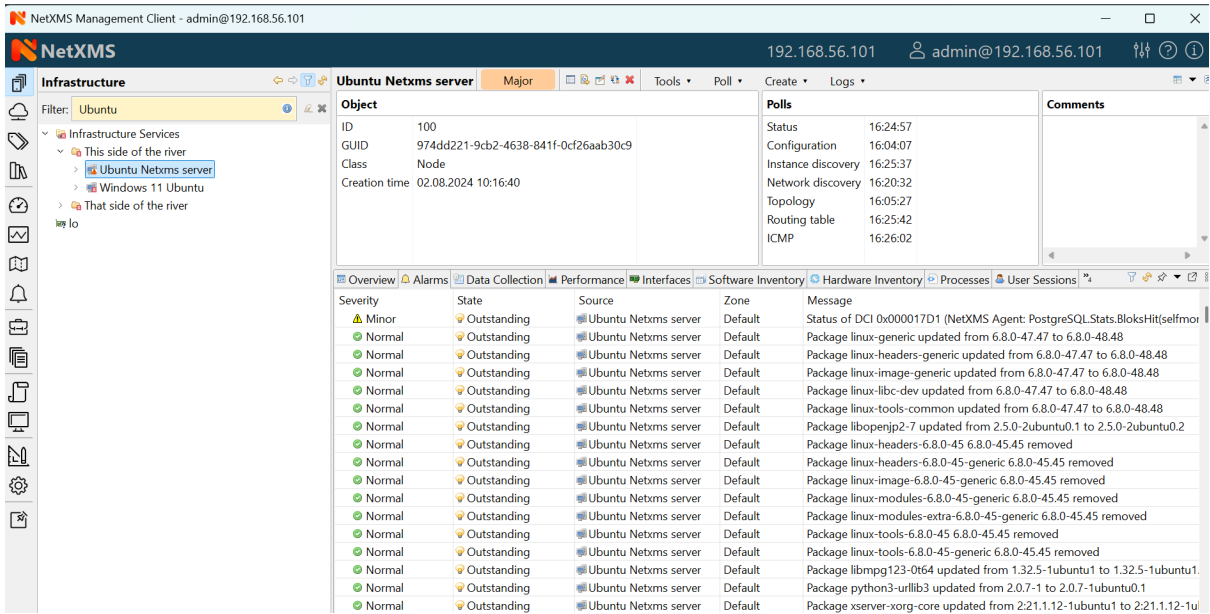Without prefix search is performed by object name.

Fig. 2: As-you-type filter in action

3.4 Object details

This view provides one or more tabs with detailed information about object currently selected in *Object Browser*. List of available tabs depends on type of the selected object.
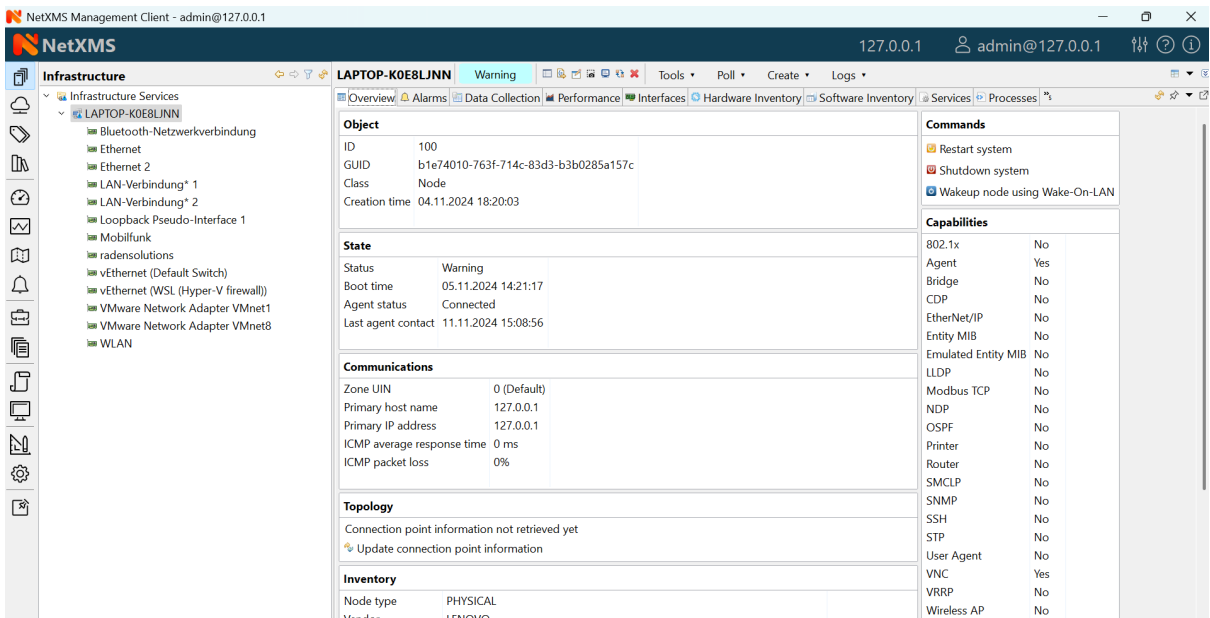
3.4.1 Overview

Fig. 3: Overview tab

This view provides basic information about selected object: Name, Class, Status and comments. For *Node* objects, it also show IP address, Host name, SNMP details as well as Capabilities.

Node capabilities

Node capabilities displays results of capability auto-detection (it's performed on Configuration Poll). E.g.:

Capability	Description
Agent	True if NetXMS Server can communicate with NetXMS agent installed on the node
Router	True if selected object can route network traffic
SNMP	True if NetXMS Server can communicate with this device via SNMP protocol
SSH	True if NetXMS Server has credentials and is able to connect to this device via ssh

3.4.2 Alarms
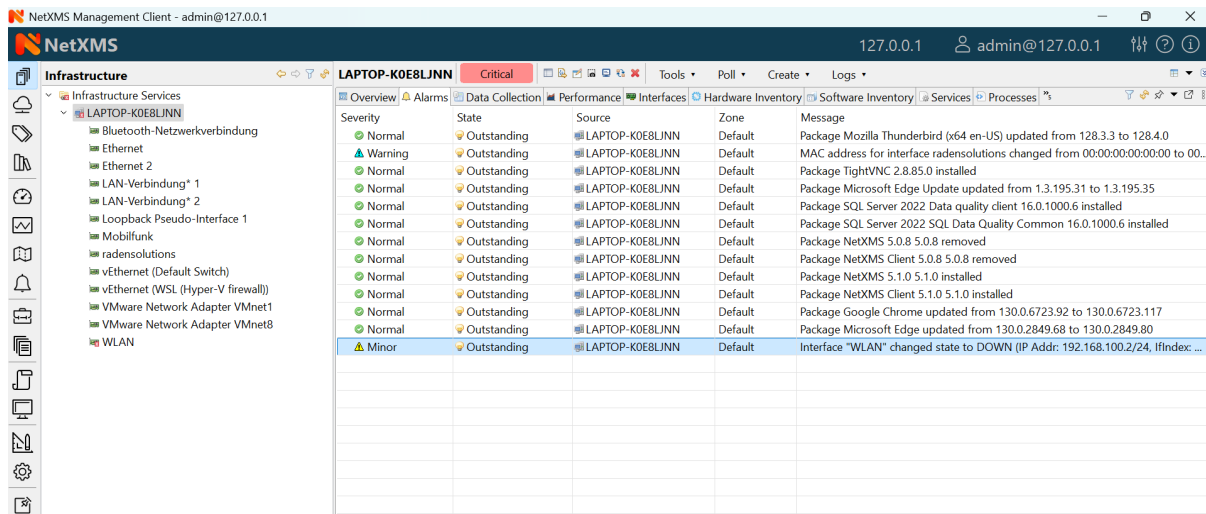
Alarm view provides user with list of alarms for currently selected element of the tree, including all child objects. To view all alarms in the system, either use system-wide *Alarm Browser* (click *View ▸ Alarm Browser* to open) or select *Entire Network* object. Right-click on the alarm will open pop-up menu with available actions
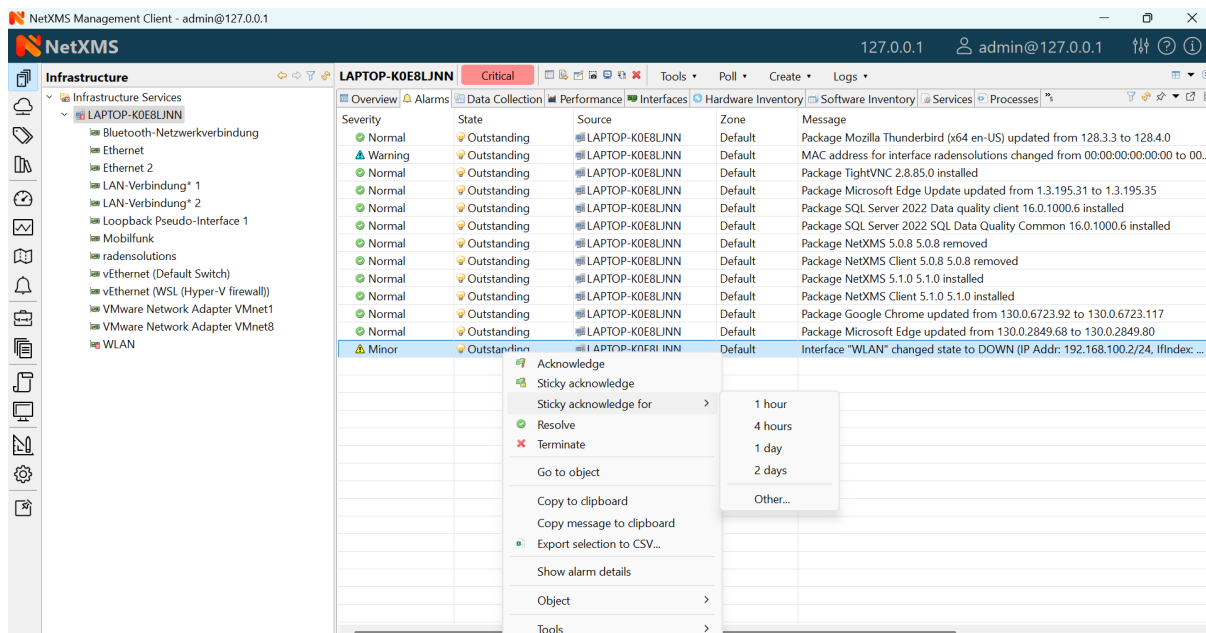

Fig. 4: Alarm context menu

Each alarm can be in one of the following states:

State	Description
Outstanding	Newly created alarm, no actions was taken by user
Acknowledged	User acknowledged raised issue, work in progress
Resolved	Issue resolved, but alarm is kept in the list. This state mostly used when alarm is automatically resolved by the system, to keep users informed about incident
Terminated	Issue resolved and alarm removed from list.

3.4.3 Data collection
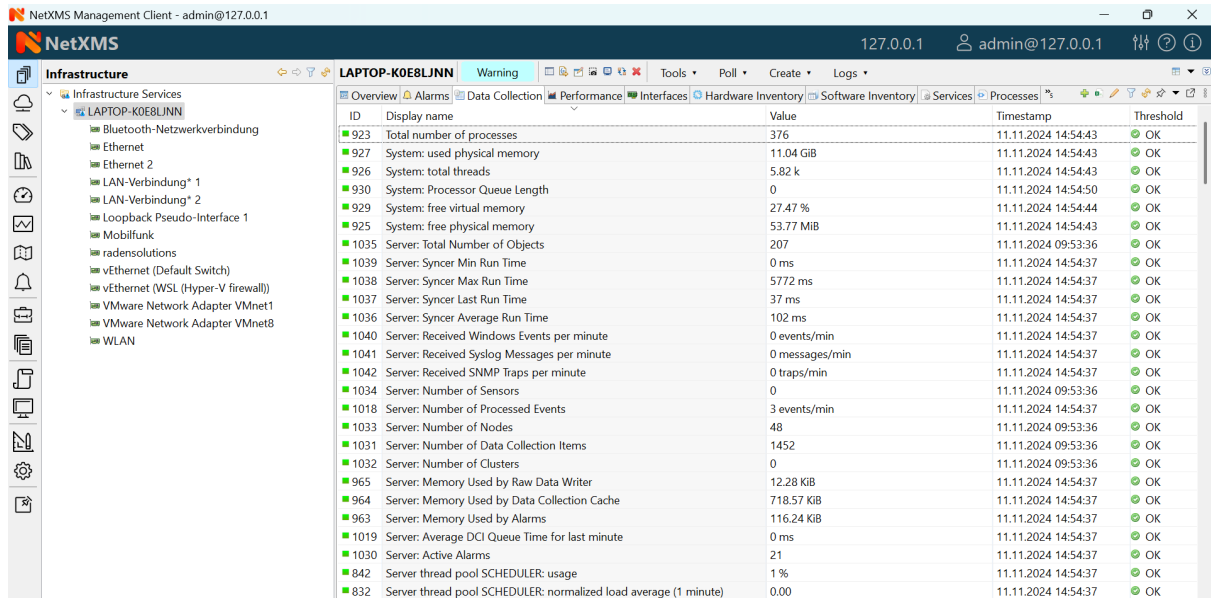
This view provides access to all collected data, both latest and historical. When view is shown, it displays latest values, as well as timestamp when each value was collected. Threshold column indicates threshold violations for given *DCI*. User has two options to interact with data:

- Double click on a *DCI* will open line graph view for last hour

- Right-click on a *DCI* will open pop-up menu giving access to available actions

    - *History* - show historical data

    - *Line Chart*, *Pie Chart*, *Bar Chart* - show historical data in graphical form

    - *Clear collected data* - remove all history for selected *DCI*

Clicking *Edit mode* button on the toolbar changes Data collection view into editing mode. In this mode information about DCI configuration is displayed, double click on a *DCI* will open it for editing.
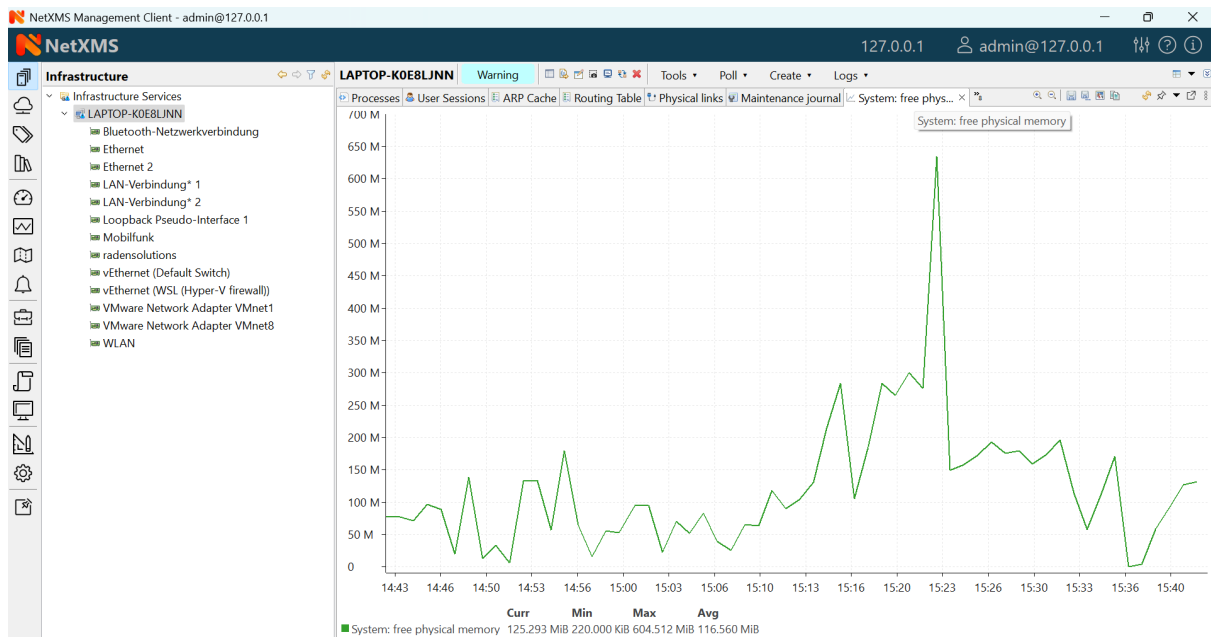
Fig. 5: Line graph built from collected data

3.4.4 Performance Tab

Performance tab is a special view that allows to quickly assess health of the selected node using one or more graphs predefined by administrator. Each graph can contain data from multiple *DCIs* on a node .
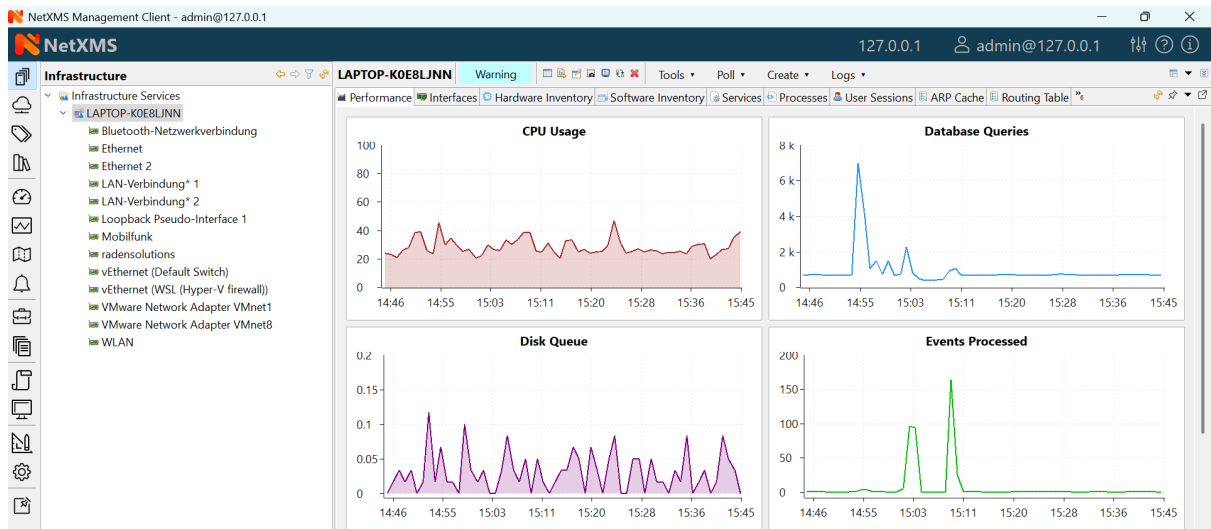

Fig. 6: Router's CPU usage displayed

3.5 Network Maps

This view allows user to see network overview in a map form. Map can be build and routed either manually or automatically for selected part of the network. Maps can be automatically generated based on:

- Layer 2 network topology

- IP (layer 3) topology

- OSPF topology
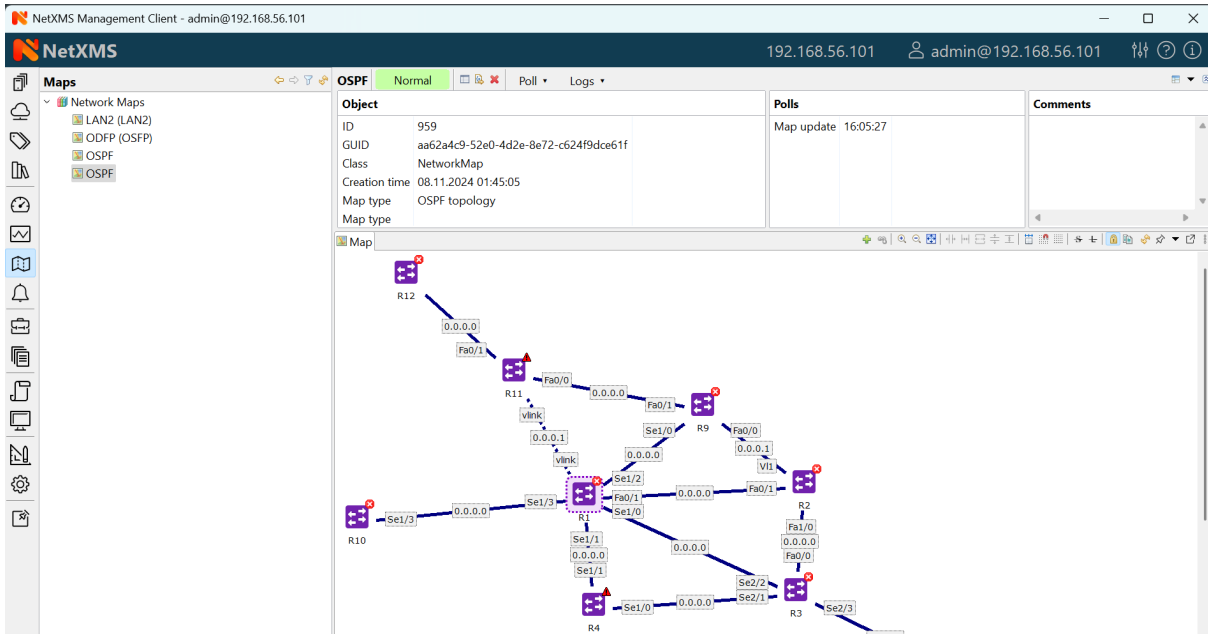
- Internal communication topology

Fig. 7: Automatically built network map based on OSPF topology

To open existing map, left-click on the name in *Object Browser*.

3.6 Reports

NetXMS is integrated with *Jasper* reporting engine from Jaspersoft. This view allows user to generate report and download result as PDF file. Report generation can take long time, so it's done in background, without user interaction. When report is generated, resulting PDF can be downloaded any time, as well as any result from previous runs.
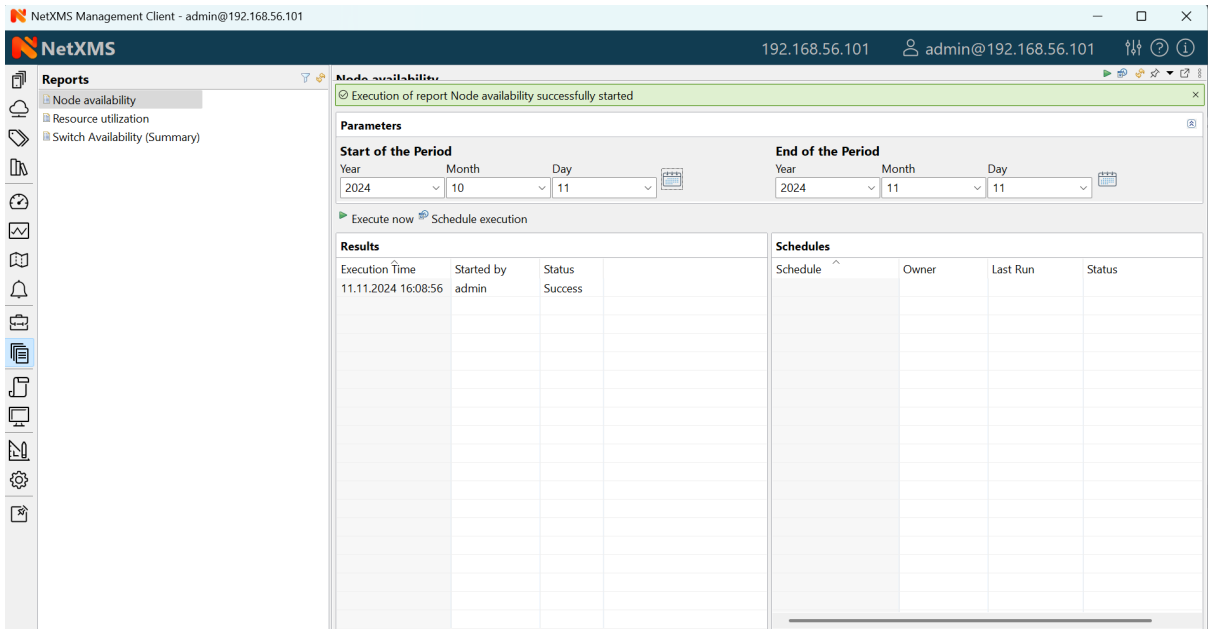
To generate report:

- Left-click on report name in *Object Browser*, report view will open (as show in figure above)

- In report view, fill parameters and click *Generate Report*

You can monitor progress in *Server Jobs* view. To open it, select *Window ‣ Show view ‣ Other ‣ Server Jobs*.

When report is generated, new finished job will appear in *Results* table of the view. Select it and click on *Render to PDF* to download.

When generated report data is not longer needed, it can be deleted from the sever by selecting job in *Results* view, and then clicking *Delete*.
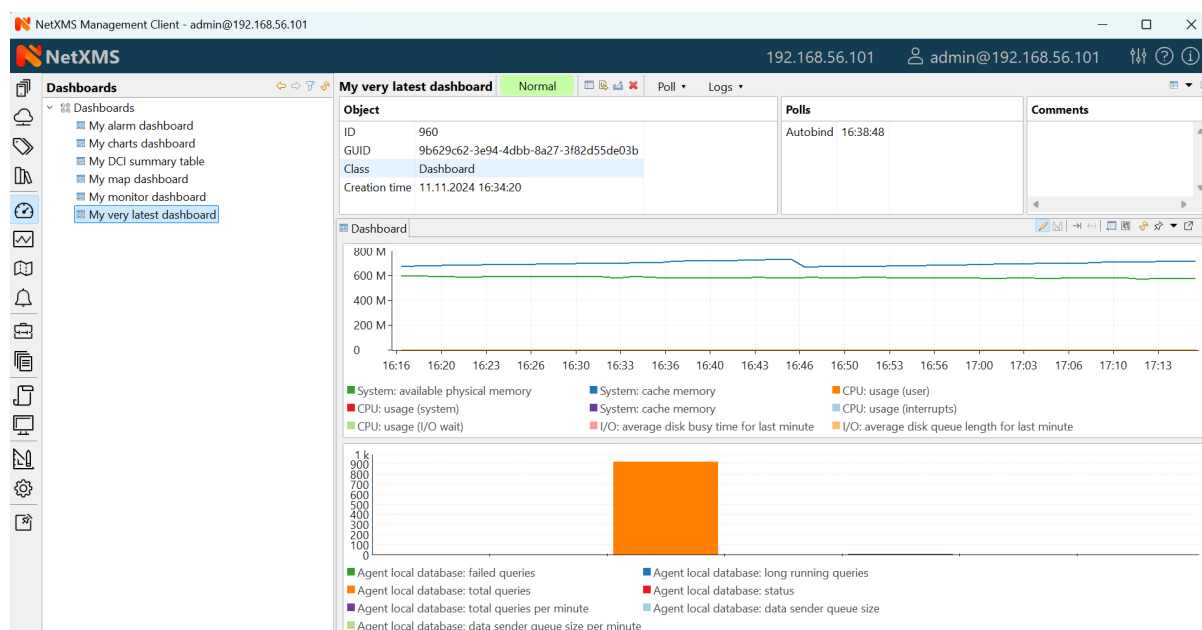
3.7 Dashboards

Fig. 8: Dashboard showing system and agent DCI readings.

Dashboards are defined by administrator and allow to combine any available visualization components with data from multiple sources in order to create high-level views to see network (or parts of it) health at a glance.

To open a dashboard, switch to *Dashboard* perspective and select dashboard with left-click.
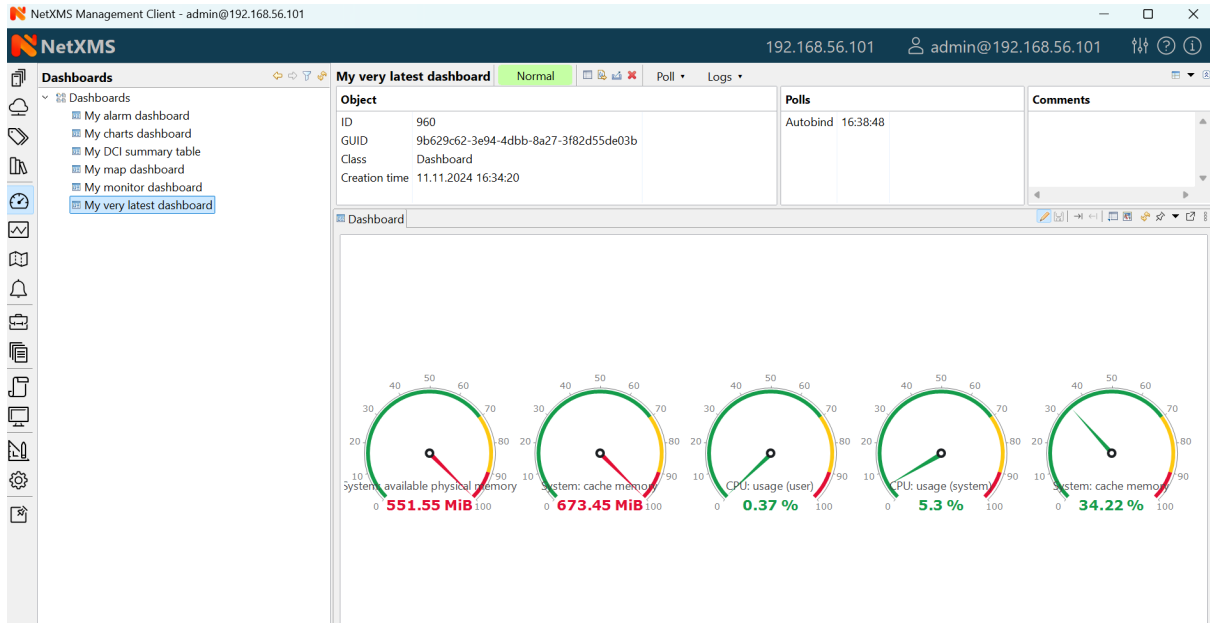
Fig. 9: Dashboards perspective

3.8 Business Services

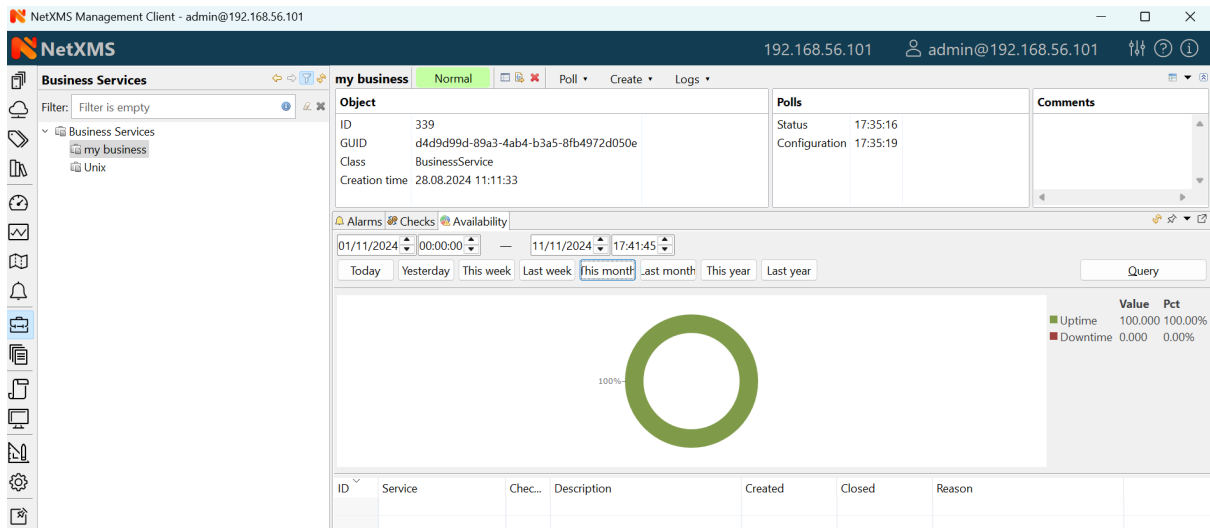Fig. 10: Availability chart and uptime percentage for a system.

Business Services is a hierarchy of logical services as defined by administrator. Each service can represent combined state of multiple elements. For each service in the hierarchy, NetXMS calculates availability percentage and keeps track of all downtime cases. To check availability of any particular level, select it in *Object Browser*.

OBJECT MANAGEMENT

4.1 Objects

Detailed information about objects, it's usage, parents and children can be found in concept chapter, *Objects*. In this section will be described only actions and properties that can be applied on different object classes.

4.1.1 Node context menu

Many of menu items described here are also available on other types of objects - sensors, collectors, racks, access points, etc.

When node is unmanaged/managed - all it's children like interfaces and service monitoring are also unmanaged/managed. In unmanaged state *metrics* are not collected and no pols are scheduled.

Node's zone can be changed using *Change zone...* menu item. *File manager* will open agent file manager view. *Upload file* can be used to upload file from server to node. This action can be applied simultaneously to multiple nodes.

Remote control allows to access remote node via VNC.

Take screenshot takes screenshot. Requires NetXMS agent to be installed on the node. Currently screenshot can be taken only from Windows machines.

Poll menu contains a list of available polls that can be performed on a node. The following options are available:

Poll Name	Description
Status	Determine current status of an object
Configuration	Determine current configuration of an object (list of interfaces, supported protocols, etc.) By default executes auto bind scripts for templates and containers, use "Objects.AutobindOnConfigurationPoll" server configuration variable to disable.
Configuration (full)	Same as usual configuration poll but resets previously detected capabilities and detects them again. (can only be executed manually)
Instance discovery	Perform Instance Discovery to add/remove DCIs
Topology	Gather information related to network link layer topology

Under *Tools* menu are available predefined object tools that will be executed on selected node. More about object tool configuration can be found there: *Object Tools*.

Logs menu provides access available logs.

Execute script... opens NXSL (built-in scripting language) execution view.

Topology maps menu gives options to build adhoc topology maps based on Layer 2, IP and Internal Connection topology.

Route to... will build network map with route to selected node from node that is selected from Object selector window.

Route from... will build network map with route from selected node to node that is selected from Object selector window.

Route from NetXMS Server will build network map with route from NetXMS server to selected node.

MIB Explorer (available only on SNMP-capable nodes) opens MIB explorer view that allows walking SNMP OIDs and reading information from MIB files.

Change zone… allows to change zone of selected node.

4.1.2 Subnet, Container and Collector context menu

Some actions, performed on objects, whose children are nodes (sensors, access points, etc) are executed on these nodes and not on object where it was called. These actions are:

Manage / *Unmanage*. Management status will be applied to all nodes under subnet or container.

Upload file menu item will upload file from server to all nodes that have agent.

Under *Tools* menu are available predefined object tools that will be executed on each subnet node. More about object tool configuration can be found there: *Object Tools*.

Alarms menu item will open view with all subnet nodes' alarms.

If subnet or container is deleted and is the only parent of a node, then node also will be deleted with the subnet.

4.1.3 Condition

Conditions may represent more complicated status checks because each condition can have a script attached. Interval for evaluation of condition status is configured in Server Configuration Variables as ConditionPollingInterval with default value 60 seconds. Input values for the condition script can be set in object properties. Such values are accessible via $1, $2, … variables inside the script. If the script returns 0, an activation event with the defined severity is created. If the script returns any other value, then a deactivation event is created.

Condition can be manged/unmanaged. If condition is unmanaged, evaluation of condition is not executed.

4.1.4 Container

Everything described in this chapter is also related to collectors, which are basically containers with data collection capabilities.

Containers can be created in Infrastructure Services tree. Existing nodes and subnets can be added to containers by using Bind operation, and removed by using Unbind operation. New nodes, conditions, clusters, containers, collectors, sensors and racks can also be created. They can be created using required menu item of container under which this object should appear. Containers and nodes inside them can be moved by *Move to another container* menu item or using drag&drop.

Menu items:

There are special menu item for each object that can be created in container. Objects like rack, container, mobile device, cluster are manually created objects. Node can be manually created or found by network discovery. In case if it is required to add already existing object to container use *Bind…* menu item. To remove node from container, but do not delete it use *Unbind…* menu item.

4.2 Object Tools

Object tools are configured in NetXMS settings for executed on objects. Tools are shown under "Tools" item of node menu. There are some pre defined object tools, they can be disabled and new ones can be configured by NetXMS administrator.

NETWORK TOPOLOGY

5.1 Introduction

NetXMS server automatically creates and maintains network model on different layers. All necessary information taken from ARP cache, routing tables, and switch forwarding database of managed nodes. Topology data provided by CDP, LLDP, and NDP (SONMP) protocols also used in building network model. Having network model instantly available allows NetXMS users to perform various network topology tasks much faster and easier.

Requirements to build network topology:

- All network equipment should be registered in NetXMS system

- Equipment should response to SNMP

- Equipment should have at least STP

- There will be more information if equipment will have LLDP or CDP

Manual topology poll can be started on the network equipment to heave information about information availability.

Based on network topology network correlation is done. Network correlation reduce number of alerts and increase problem resolution speed.

Currently there are 3 states/events regarding connectivity:

- down (event SYS_NODE_DOWN) - when server cannot contact the node and has no topology information for event correlation or it is really problem with that node

- unreachable (SYS_NODE_UNREACHABLE) - when server knows that node cannot be contacted due to intermediate router/interface failure

- up (SYS_NODE_UP) - when node is reachable

So when node becomes unreachable, either SYS_NODE_DOWN or SYS_NODE_UNREACHABLE event is generated, depending on root cause. But when node became reachable again, SYS_NODE_UP being generated.

5.2 How topology information built

FDB. From FDB table we take ports where only one mac address is present - this means that something is directly connected. If this device is present in NetXMS and it's mac address is known (we have agent on it, SNMP, or some other agent on that network communicated to that device and has IP-MAC pair in ARP table) - we have a peer.

LLDP. So if we have another switch connected, that switch is sending LLDP packets, the switch that we are polling receives these packets and saves information in LLDP table. We read this table and we know that there's a device with some LLDP ID connected to port X of our device. But we also need NetXMS to read that device via SNMP, in this case LLDP ID will be read and we will be able to match.
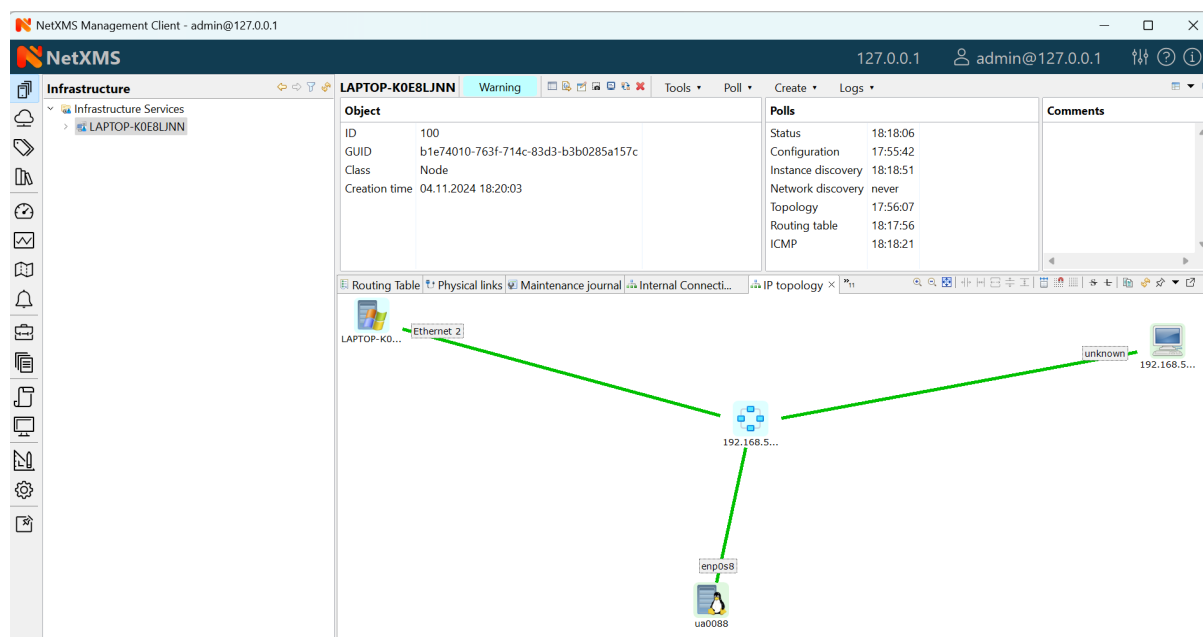
CDP. Similar to LLDP.

STP table on a switch has limited information - only about peers that are on the way to root LLDP switch. But we read that and can get peers from there.

Interfaces tab has Peer Discovery Protocol` column which tells, how peer information was obtained.

For debug you can set debug tags poll.topology, topo.*, topology.* to level 7 - there will be some information in server log when topology poll is executed.

5.3 Find where node is connected

It is possible to find switch port where any given node is connected (sometimes called "connection point" in management client). To find out node's connection point, right-click on node object, and select Topology Map. Further three option will be available (Layer 2 topology, IP topology and Internal communication topology). Here is IP topology example:

Columns have the following meaning:

Seq.	Search result sequence number
Node	Name of end node object
Interface	Name of node's interface object
MAC	Interface's MAC address
IP	Interface's IP address
Switch	Name of switch node object
Port	Name of interface object representing switch port
Type	Connection type - direct or indirect. Direct connection type means that NetXMS server did not detect any other devices on sdame switch port, and most likely end node connected directly to the switch. Indirect means that some other devices was detected on same switch port. Virtual machines and virtual machine host will always be detected as indirect.

5.4 MAC address search

It is possible to find location of any known MAC address in the network. To do this, select *Tools ‣ MAC address search*. Results of a search will be displayed in the same results view. It is not necessary that node with given MAC address be managed by NetXMS server, but if it is, appropriate details will be displayed.

5.5 IP address search

It is possible to find location of any known IP address in the network. To do this, select *Tools ▸ IP address search*. Results of a search will be displayed in the same results view. It is not necessary that node with given IP address be managed by NetXMS server, but if it is, appropriate details will be displayed.

GLOSSARY

Alarm Browser
> View, which shows all active alarms in the system and provides tools to interact with them

DCI
> Data Collection Item, configuration element, which contains parameter to collect (for example "CPU Usage"), collection schedule and thresholds

Entire Network
> Automatically generated hierarchy that contains all nodes known to NetXMS

Metric
> One entity of collected data

Node
> Object that represents physical server

Object
> Representation of logical or physical entity.

View
> A portion of application window displaying some information and allowing user interaction

View Stack
> Multiple views combined into single one, with tab navigation on top of it