# VISULOX 5

## Admin Guide - VLX Portal

15.06.2026

visulox

# Table of Contents

1 VISULOX NativeClient and VispraClient ..... 5
2 VISULOX Portal Architecture ..... 6
3 VISULOX Dashboard ..... 15
4 Monitoring & Log Inspection Guide ..... 66
5 Database Backup Guide ..... 76
6 Health Check for VISULOX Portal ..... 82

## Limitation of Liability Statement (Disclaimer)

The Visulox documentation is proprietary to amitego engineering GmbH and no ownership rights are hereby transferred. No part of the documentation shall be used, reproduced, translated, converted, adapted, stored in a retrieval system, communicated or transmitted by any means, for any commercial purpose, including without limitation, sale, resale, licence, rental or lease, without the prior express written consent of amitego engineering.

amitego engineering does not make any representations, warranties or guarantees, express or implied, as to the accuracy or completeness of the documentation. Users must be aware that updates and amendments will be made from time to time to the documentation. It is the user's responsibility to determine whether there have been any such updates or amendments. Neither amitego engineering nor any of its directors, officers, employees or agents shall be liable in contract, tort or in any other manner whatsoever to any person for any loss, damage, injury, liability, cost or expense of any nature, including without limitation incidental, special, direct or consequential damages arising out of or in connection with the use of the documentation.

- VISULOX NativeClient and VispraClient
- VISULOX Portal Architecture
- VISULOX Dashboard
    - Access Explorer
    - User Console
    - Application Management
    - Server Management
    - VISULOX Cluster Management
    - Access Tokens for external access to the VISULOX Dashoard
- Monitoring & Log Inspection Guide
- Database Backup Guide
- Health Check for VISULOX Portal

# 1 VISULOX NativeClient and VispraClient

With the development of VISULOX 5, we set ourselves a clear goal: deliver a **fast and seamless experience** when users work with applications through the VISULOX client.

To achieve this, we **Vispra** has been developed.

## 1.1 Vispra Client

Vispra is VISULOX's lightweight, high-performance client — built from the ground up for speed and simplicity.

Vispra is available in two variants:

- **Web Client** — runs directly in the browser, no installation required.
- **Native Client** — a desktop application for Windows, macOS, and Linux with a small footprint and fast startup.

Both variants share the same interface and connect to the same VISULOX backend with full security policies, session management, and auditing.

### 1.1.1 Key strengths

- Lightweight and fast — minimal download size, quick installation, significantly faster session startup.
- Unified experience across web and desktop.
- Optimized performance, especially for Windows-based remote applications.
- Cross-platform foundation (Windows, macOS, Linux).

**Current limitation:** Vispra does not yet support **seamless mode** (individual application windows integrated into the local desktop).
If seamless mode is a requirement, the Xpra Client should be used instead which is available on request.

# 2 VISULOX Portal Architecture

## 2.1 VISULOX 5 Architecture

### 2.1.1 Introduction

In this document, the architecture of VISULOX 5 is described in detail. The goal of this document is to explain the different components of the system, how they are defined, how they interact with each other, and what responsibilities each part has.

The terminology used in this document follows specific naming conventions for each component of the system.

---

### 2.1.2 Table of Contents


1. VISULOX Core
2. VISULOX Portal
3. VISULOX Portal Connector
4. VISULOX Portal AppLauncher
5. Node Gateway
6. Main Gateway
7. Conclusion


---

### 2.1.3 1. VISULOX Core

VISULOX Core is defined as the central brain responsible for:

- Managing policies
- Recording sessions
- Managing session lifecycle

This component is written in **TCL** and is essentially a modified version of VISULOX 4.

---

### 2.1.4 2. VISULOX Portal

VISULOX Portal is the main business logic controller responsible for defining and managing:

- User Profiles
- Applications
- Application Servers
- Relationships between them

#### 2.1 Dependencies

For the system to start, the following dependencies must be available:

- **PostgreSQL:** The central database used to store system data.
- **KEYDB:** A key-value database used for message brokering and data storage.

#### 2.2 Services Provided:

- **Client WebSocket:** Each native client (VISULOX Client Wrapper), which runs XPRA on the user's machine, connects to VISULOX Portal through this service.
- **UI4 WebSocket:** Each browser client that logs into VISULOX via a browser requires UI updates (UI4), which are delivered through this connection.
- **SOAP APIs:** All requests sent from VISULOX Core are processed via SOAP protocol by VISULOX Portal.
- **Dashboard APIs:** All APIs used in the Portal Console dashboard are provided by VISULOX Portal. *These APIs are protected using a cookie set during login.*
- **Internal APIs:** All APIs required by internal subsystems (such as VISULOX Portal Connector or other Portal servers) are provided here. *These APIs are protected using JWT*. Each component has a defined secret and receives an Access Token and Refresh Token during a handshake process.
- **Admin APIs:** All APIs required by administrators for programmatic interaction with the system. *These APIs are similar to Dashboard APIs but differ in authentication and authorization mechanisms.*

### 2.3 Important Notes

⚠ VISULOX Portal acts as a **control unit**.
It makes decisions and sends commands to other components but does not perform execution itself.
For example, it does NOT:

- Create Frame Buffers
- Control the lifecycle of Frame Buffers
- Configure NGINX
- Directly communicate with VISULOX Core APIs

**The system is designed so that VISULOX Portal can run all services listed in section 2.2 or only a subset of them.**

⚠ In host environments (bare-metal or VM), it is possible to install databases alongside VISULOX servers.
For example, in a cluster with 5 servers:

- 2 servers may run VISULOX + databases
- 3 servers may run VISULOX only

In this context:

- **Master (master0, master1)** → servers with VISULOX + databases
- **Worker** → servers with only VISULOX

⚠ In cloud environments, the concept of master/worker does not exist.
All servers are treated equally.

⚠ In VISULOX 5 architecture, databases are considered **infrastructure**, not part of the core product.
Therefore, customers are free to configure them as they prefer.

## 2.1.5 3. VISULOX Portal Connector

VISULOX Portal Connector acts as the **execution arm** of VISULOX Portal.

Since VISULOX v5 consists of multiple components (*VISULOX Core, NGINX, XPRA*), each using different protocols:

- VISULOX Core → ZMQ + REST
- NGINX → configuration files
- XPRA → CLI APIs

The responsibility of the Connector is to implement communication with each component and execute Portal commands accordingly. Also, if a new component is added or replaced in future versions of the software, it is the responsibility of the Connector to build the bridge between the Portal and that component.

## 3.1 Dependencies

- **Portal Address:** Address of at least one VISULOX Portal server
- **Secret:** A defined secret for communication

## 3.2 Services Provided

- **RESTful API:** Executes all commands received from VISULOX Portal, including operations on XPRA, NGINX, and VISULOX Core.
- **XPRA Integration:**
  - Applies Portal commands to XPRA
  - Manages XPRA sessions
  - Performs cleanup **Emulator Sessions**
  - Works with AppLauncher for RDP/SSH execution
- **VISULOX Core Integration:** Executes Portal commands on VISULOX Core.
- **NGINX Integration:** Applies configuration changes to NGINX (Main Gateway) and updates it.
- **File Upload Handling:** Handles uploads using the Tus protocol plugin.
- **Entra Login Integration:** Handles Entra (Azure) login via plugin.

### 3.3 Notes

Connector is a binary that can run on Windows, Linux, and macOS. It can leverage the capabilities of each platform to serve the Portal.

The Connector application can provide either a single service or all services together.

If the Connector is only responsible for communication with Xpra, it is referred to as an **XPRA-Connector** mode.
If it is only responsible for communication with VISULOX Core, it is called a **VLX-Connector** mode.
And if it is only responsible for handling NGINX in **Main Gateway** mode, it is called a **Main-Gateway-Connector** mode.

- XPRA Connector mode → handles XPRA only
- VLX Connector mode → handles VISULOX Core only
- Main Gateway Connector mode → handles NGINX in Main Gateway (Section 6)

Connector is responsible for managing and monitoring Frame Buffers (FB). Based on Portal decisions, it can start, stop, or suspend them.

The Connector application manages each Frame Buffer (FB) using two mechanisms:
a) **Passive mechanism**: It checks at short time intervals whether the FB is in a healthy state and whether all windows and X11 clients are running properly.
b) **Active mechanism**: In this approach, it detects changes in the FB by attaching several webhooks to XPRA.

## 2.1.6 4. VISULOX Portal AppLauncher

AppLauncher is a Python-based utility responsible for establishing RDP and SSH connections.

### 4.1 Dependencies

- Python ≥ 3.12
- xfreerdp
- paramiko
- PythonTK

### 4.2 Services Provided

- Execute applications over RDP
- Execute applications over SSH
- Shared folder support for RDP and SSH
- SSO execution

### 4.3 Notes

AppLauncher is a sub-component of the Connector and only communicates with it.

SSH connections are currently implemented using **PExpect**.

All usernames and passwords used for SSO are transferred in encrypted form.

Each Emulator Session has its own encryption key:

- Generated by VISULOX Portal
- Only accessible to AppLauncher

- NOT accessible to Connector

### 2.1.7 5. Node Gateway

Node Gateway refers to an NGINX instance running on a host.

In a host environment (non-cloud), each VISULOX server typically includes:

- VISULOX Core
- VISULOX Portal
- VISULOX Portal Connector
- XPRA
- NGINX
- AppLauncher

The role of Node Gateway is to route traffic internally between these components.

#### 5.1 Dependencies

- NGINX configuration files

#### 5.2 Services Provided

- Routing requests to internal services
- Serving static files (HTML, JSON, CSS)
- Managing service ports
- Providing TLS

### 5.3 Notes

⚠ Node Gateway is designed for host environments (bare-metal or VM).
In cloud environments, its functionality can be replaced by:

- Kubernetes Service
- Kubernetes Ingress

NGINX configurations are designed so administrators can deploy additional NGINX servers if needed.

---

## 2.1.8 6. Main Gateway

Main Gateway is an NGINX instance deployed at the **edge (internet-facing layer)**.

Its main responsibilities are:

- Load balancing
- Sticky connections

### 6.1 Dependencies

- NGINX configuration files (can be updated by Connector)

### 6.2 Services Provided

- **Load balancing** across servers
- **Sticky connections**: Due to VISULOX Core architecture, after login (UI4 panel), user requests must always be routed to the same node. Main Gateway ensures session affinity.
- **Automatic NGINX configuration updates** (via Main Gateway Connector)

### 6.3 Notes

⚠ Used in host environments with clustering.
In cloud environments, this is typically handled automatically.

A Connector runs alongside NGINX to update its configuration dynamically.

⚠ If running in cloud or without clustering, Main Gateway is not required.
Node Gateway alone is sufficient.

⚠ **As of version 0.9.3, Main Gateway and Node Gateway cannot run on the same server.**

## 2.1.9 Conclusion

VISULOX 5 is designed as a modular system with clear separation of responsibilities between components.

- VISULOX Portal → decision making
- Connector → execution
- Core → session and policy engine
- AppLauncher → remote execution layer
- NGINX → routing and traffic management

This architecture allows:

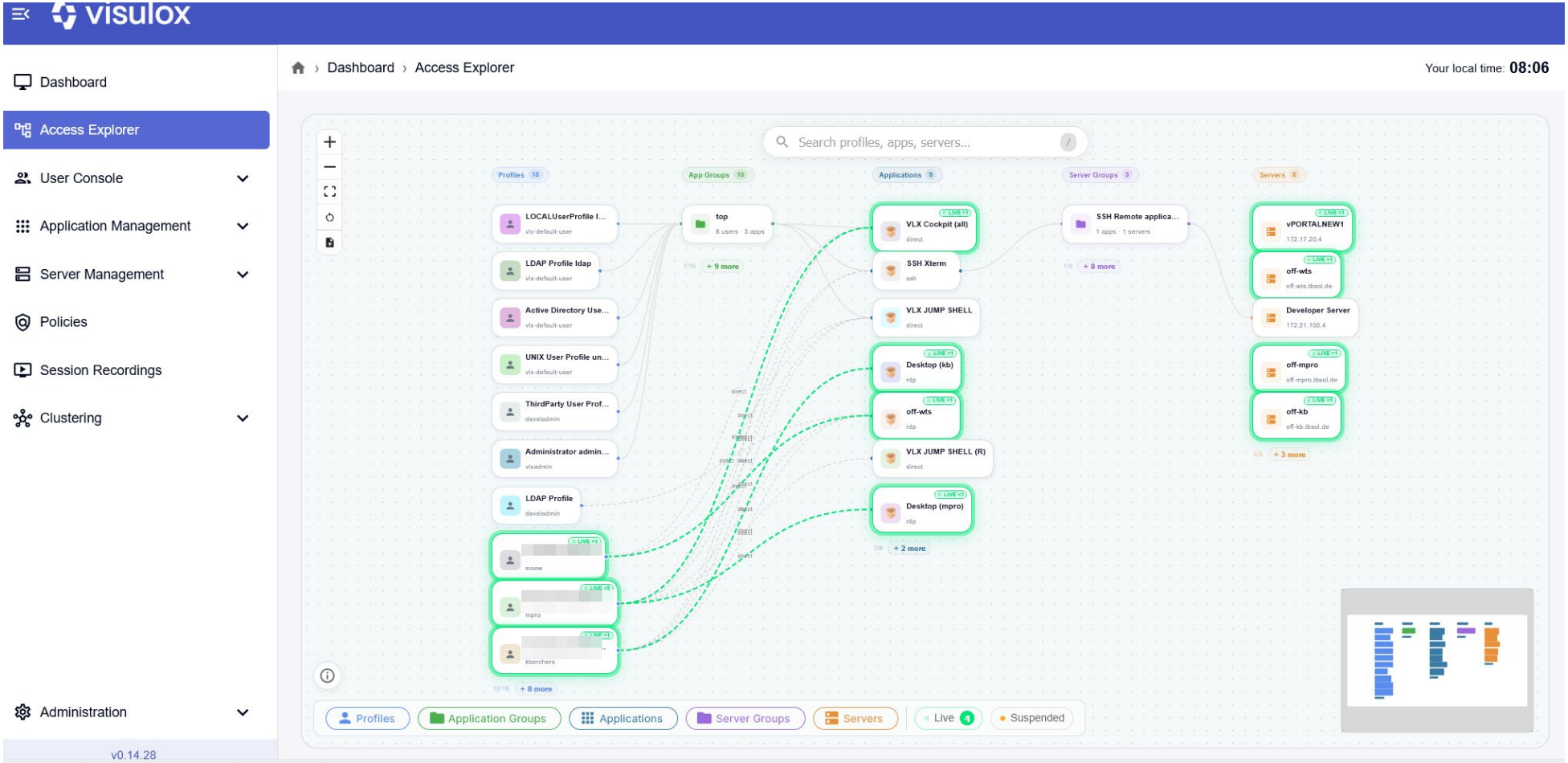
- Flexible deployment (host or cloud)
- Scalability across clusters
- Clear separation between infrastructure and product logic

# 3 VISULOX Dashboard


- Access Explorer
- User Console
- Application Management
- Server Management
- VISULOX Cluster Management
- Access Tokens for external access to the VISULOX Dashoard


## 3.1 Access Explorer

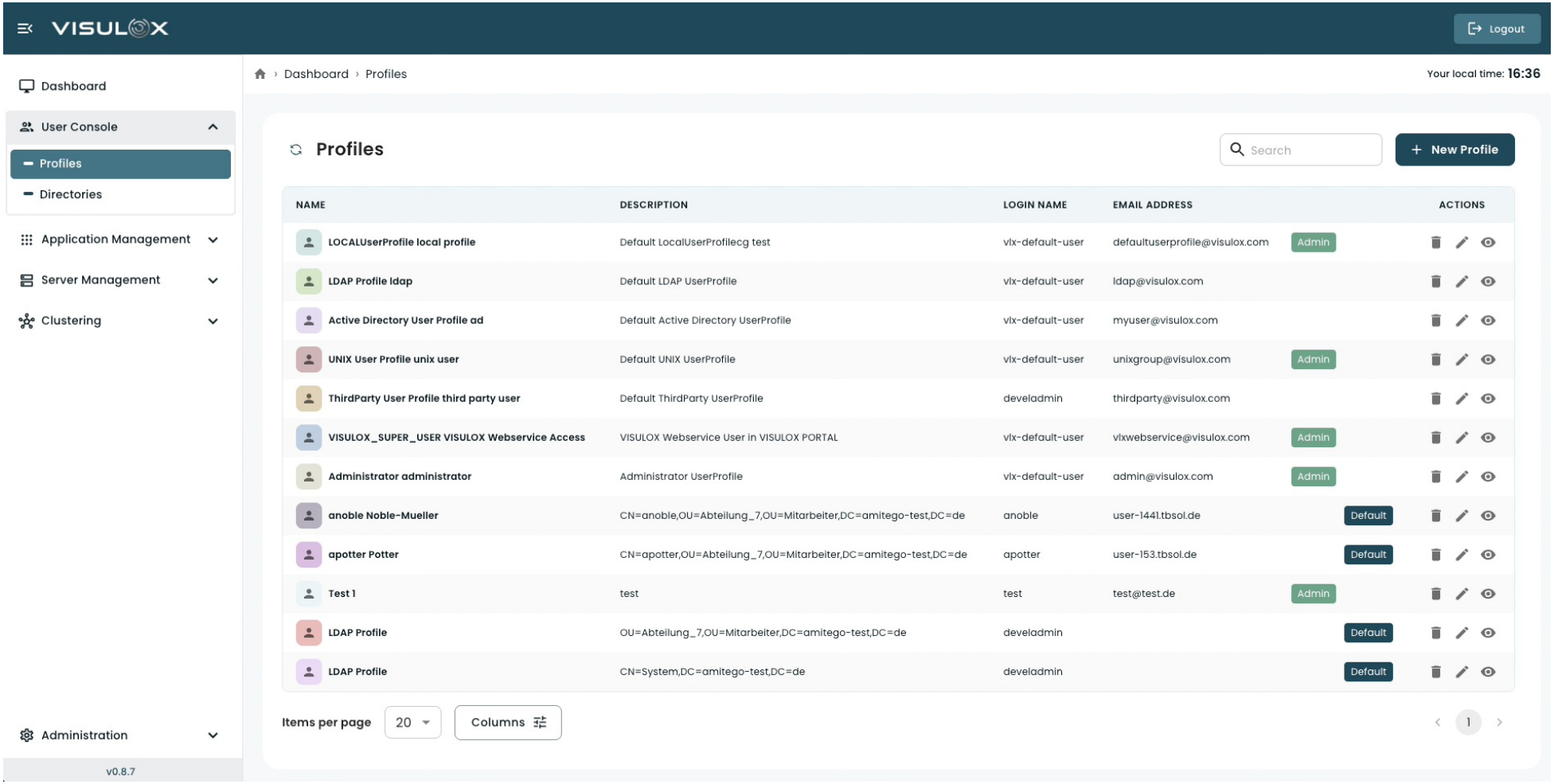
The Access Explorer in Dashboard displays all profiles / users and their assigned applications, servers.
Live sessions are highlighted, searching objects is possible and can be filtered.


visulox
Dashboard
Access Explorer
User Console
Application Management
Server Management
Policies
Session Recordings
Clustering
Administration
v0.14.28
Dashboard › Access Explorer
Your local time: 08:06
Search profiles, apps, servers...
Profiles
App Groups
Applications
Server Groups
Servers
LOCALUserProfile l...
LDAP Profile ldap
Active Directory Use...
UNIX User Profile un...
ThirdParty User Prof...
Administrator admin...
LDAP Profile
top
+ 9 more
VLX Cockpit (all)
SSH Xterm
VLX JUMP SHELL
Desktop (kb)
off-wts
VLX JUMP SHELL (R)
Desktop (mpro)
+ 2 more
SSH Remote applica...
+ 8 more
vPORTALNEW1
off-wts
Developer Server
off-mpro
off-kb
+ 3 more
+ 8 more
Profiles
Application Groups
Applications
Server Groups
Servers
Live
Suspended

## 3.2 User Console

- User Console > Profiles
- User Console > Directories

### 3.2.1 User Console > Profiles


VISULOX
Logout
Dashboard
User Console
Profiles
Directories
Application Management
Server Management
Clustering
Administration
v0.8.7
Dashboard › Profiles
Your local time: 16:36
Profiles
Search
New Profile
NAME
DESCRIPTION
LOGIN NAME
EMAIL ADDRESS
ACTIONS
LOCALUserProfile local profile
Default LocalUserProfilecg test
vlx-default-user
defaultuserprofile@visulox.com
Admin
LDAP Profile ldap
Default LDAP UserProfile
vlx-default-user
ldap@visulox.com
Active Directory User Profile ad
Default Active Directory UserProfile
vlx-default-user
myuser@visulox.com
UNIX User Profile unix user
Default UNIX UserProfile
vlx-default-user
unixgroup@visulox.com
Admin
ThirdParty User Profile third party user
Default ThirdParty UserProfile
develadmin
thirdparty@visulox.com
VISULOX_SUPER_USER VISULOX Webservice Access
VISULOX Webservice User in VISULOX PORTAL
vlx-default-user
vlxwebservice@visulox.com
Admin
Administrator administrator
Administrator UserProfile
vlx-default-user
admin@visulox.com
Admin
anoble Noble-Mueller
CN=anoble,OU=Abteilung_7,OU=Mitarbeiter,DC=amitego-test,DC=de
anoble
user-1441.tbsol.de
Default
apotter Potter
CN=apotter,OU=Abteilung_7,OU=Mitarbeiter,DC=amitego-test,DC=de
apotter
user-153.tbsol.de
Default
Test 1
test
test
test@test.de
Admin
LDAP Profile
OU=Abteilung_7,OU=Mitarbeiter,DC=amitego-test,DC=de
develadmin
Default
LDAP Profile
CN=System,DC=amitego-test,DC=de
develadmin
Default
Items per page
20
Columns
1

## What is a Profile?

A **profile** defines what a user is allowed to access in VISULOX.

Profiles determine:

- which applications are visible after login
- what permissions the user has
- whether the user has administrative privileges
- how authentication is handled (e.g., local, LDAP, Active Directory, etc.)

Users are assigned to profiles.
Access is granted through profiles — not directly to individual users.

This approach enables centralized and scalable access management.

## Profiles Page Overview

The profiles page displays all configured user profiles.

Each entry includes key information such as:

- Profile name
- Description
- Login name
- Email address
- Role indicators (e.g., Admin, Default)

From this page, administrators can:

- Create new profiles
- Review profile details
- Modify profile settings
- Delete profiles
- Search and filter entries

- Navigate using pagination

Role indicators such as **Admin** or **Default** help quickly identify special access levels.

## Managing Profiles

### Creating a New Profile

1. Click **+ New Profile** (top-right corner).
2. Enter the required user and access details.
3. Configure permissions and application assignments.
4. Save the profile.

After saving, the profile can be assigned to users or groups.

### Viewing a Profile

Click the **View** icon to review the profile configuration in read-only mode.

### Editing a Profile

Click the **Edit** icon to update user information, permissions, or assigned applications.

Changes take effect after saving.

### Deleting a Profile

Click the **Delete** icon to permanently remove the profile.

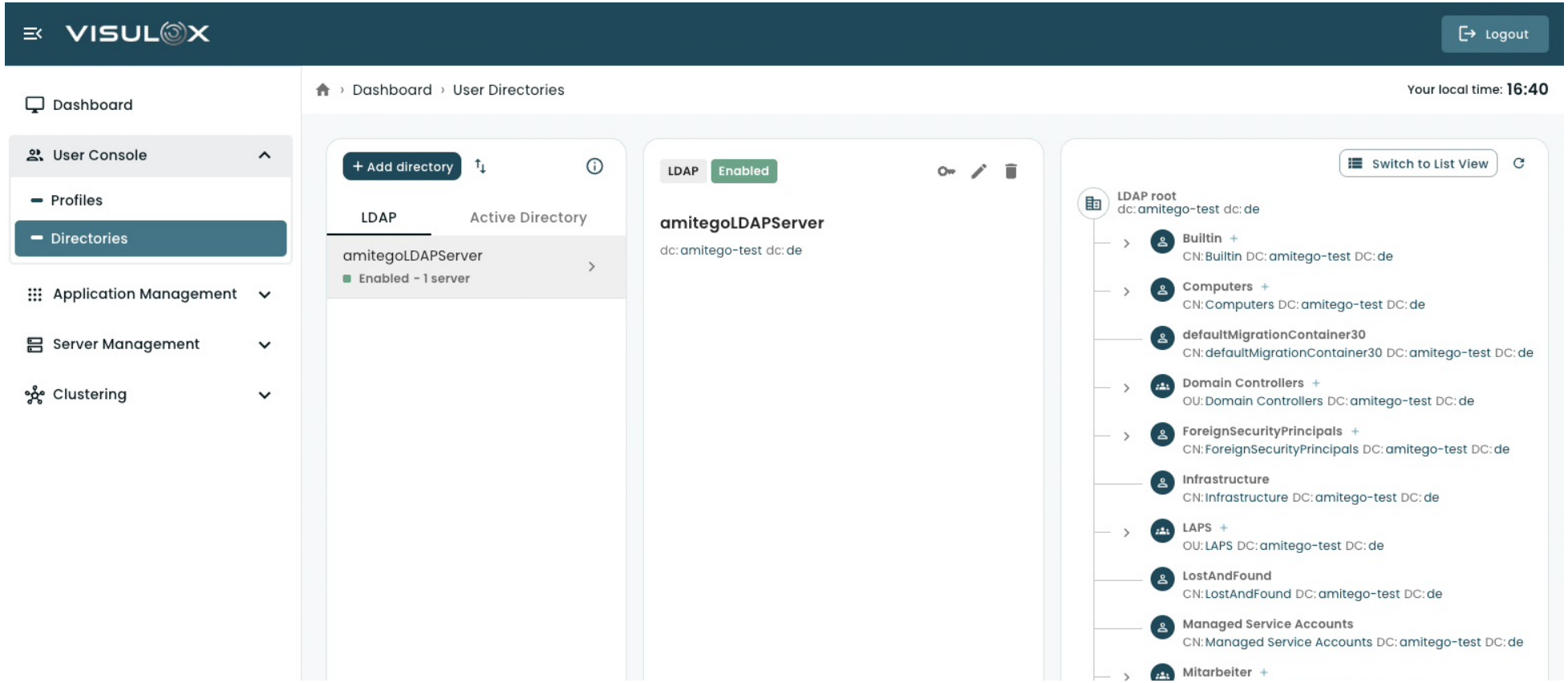
⚠ Deleting a profile removes associated access rights for users assigned to it.

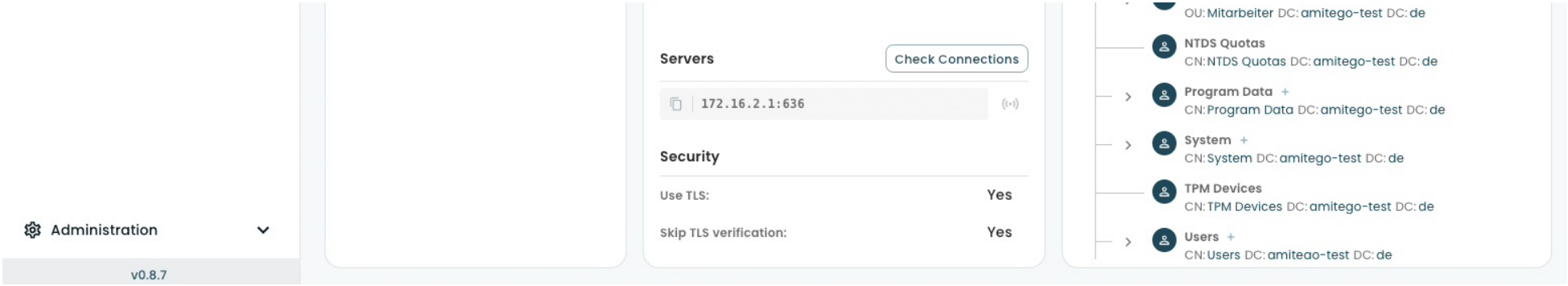
## Search and Table Controls

- Use the search field (top-right) to filter profiles by name or login.

- Use pagination controls at the bottom to navigate between pages.
- Adjust the number of items per page as needed.
- Use the column selector (if available) to customize visible columns.

### 3.2.2 User Console > Directories






## What is a Directory?

A **directory** connects VISULOX to an external identity provider.

Directories allow VISULOX to:

- authenticate users against LDAP or Active Directory
- import user accounts and groups
- synchronize organizational structures
- centralize identity management

Instead of creating local users manually, administrators can integrate an existing directory service.

## Directory Types

VISULOX supports directory types such as:

- **LDAP**
- **Active Directory**

Each directory configuration represents a connection to a specific directory server.

## Directories Page Overview

The Directories page consists of three main areas:

### 1🟦 Directory List (Left Panel)

Displays configured directory connections.

From here, administrators can:

- add a new directory
- select an existing directory
- enable or disable directory connections

### 2🟦 Directory Configuration (Center Panel)

Shows detailed configuration for the selected directory, including:

- directory name
- status (enabled / disabled)
- root DN
- connected servers
- security settings (TLS, certificate verification)

Administrators can also test connectivity using **Check Connections**.

### 3🟦 Directory Structure (Right Panel)

Displays the directory tree structure.

This allows administrators to:

- browse organizational units
- view users and groups
- confirm correct base DN configuration

- verify directory visibility

You can switch between different views if supported (e.g., tree or list view).

## Managing Directories

### Adding a New Directory

1. Click **+ Add Directory**.
2. Select the directory type (LDAP or Active Directory).
3. Enter server details and credentials.
4. Configure security settings.
5. Save the configuration.

After saving, enable the directory to activate authentication.

### Editing a Directory

Select the directory and click the **Edit** icon.

You can update:

- Server address
- Port
- Base DN
- Bind credentials
- TLS settings

Changes take effect after saving.

### Testing the Connection

Use the **Check Connections** button to verify:

- Server reachability

- Credential validity
- Proper LDAP binding

Connection testing helps diagnose configuration issues.

#### Deleting a Directory

Click the **Delete** icon to remove the directory configuration.

⚠ Removing a directory may affect user authentication if users depend on that source.

### Security Settings

Security options typically include:

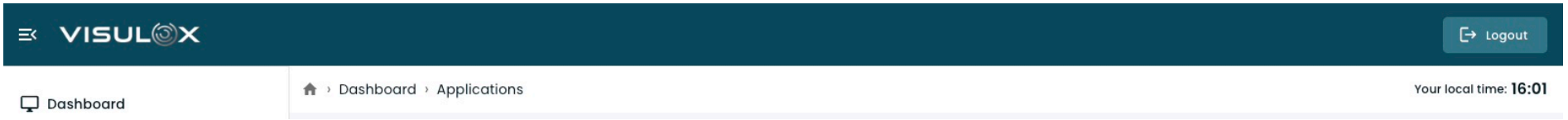
- **Use TLS** → Encrypts communication with the directory server
- **Skip TLS verification** → Disables certificate validation (not recommended for production)

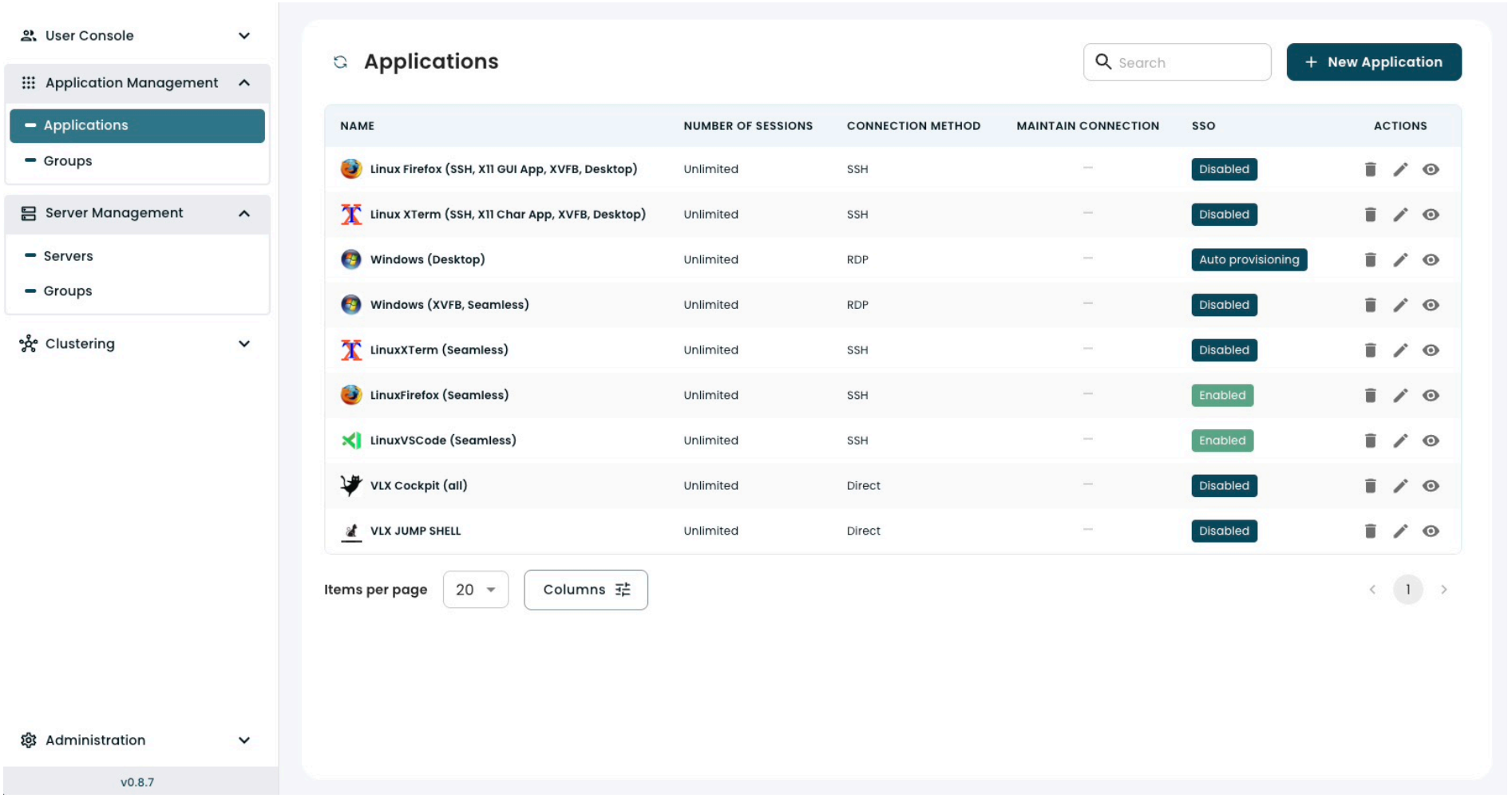
Always configure secure connections in production environments.

## 3.3 Application Management

- Application Management > Applications
- Application Management > Groups
- Application Fields

### 3.3.1 Application Management > Applications




User Console
Application Management
Applications
Groups
Server Management
Servers
Groups
Clustering
Administration
v0.8.7
Applications
Search
New Application
NAME
NUMBER OF SESSIONS
CONNECTION METHOD
MAINTAIN CONNECTION
SSO
ACTIONS
Linux Firefox (SSH, X11 GUI App, XVFB, Desktop)
Unlimited
SSH
Disabled
Linux XTerm (SSH, X11 Char App, XVFB, Desktop)
Unlimited
SSH
Disabled
Windows (Desktop)
Unlimited
RDP
Auto provisioning
Windows (XVFB, Seamless)
Unlimited
RDP
Disabled
LinuxXTerm (Seamless)
Unlimited
SSH
Disabled
LinuxFirefox (Seamless)
Unlimited
SSH
Enabled
LinuxVSCode (Seamless)
Unlimited
SSH
Enabled
VLX Cockpit (all)
Unlimited
Direct
Disabled
VLX JUMP SHELL
Unlimited
Direct
Disabled
Items per page
20
Columns
1

## What is an Application?

In VISULOX, an **application** is a launchable resource that users can access after logging in.

An application defines:

- what resource is opened
- how the connection is established
- how sessions are handled
- whether Single Sign-On (SSO) is used
- how many users can run it simultaneously

An application by itself does **not** grant access.
Access is controlled through **profiles**, which determine which applications are visible to users.

This structure ensures centralized and scalable access management.

**See also:** Application Fields.

## Applications Page Overview

The applications page provides a centralized overview of all configured applications in the system.

From this page, administrators can:

- create new applications
- review application settings
- modify existing configurations
- remove applications
- search and filter entries
- navigate through the list using pagination

Each application is displayed with key configuration indicators such as session limits, connection method, and SSO status.

## Managing Applications

### Creating a New Application

1. Click **+ New Application** (top-right corner).
2. Enter the required configuration details.
3. Save the application.

After creation, the application must be added to a **Profile** before users can access it.

### Viewing an Application

Click the **View** icon in the Actions column to open the application details in read-only mode.

### Editing an Application

Click the **Edit** icon to modify the application settings.

Changes take effect after saving.

### Deleting an Application

Click the **Delete** icon to permanently remove the application.

⚠ Deleting an application removes it from all associated profiles.

## Search and Table Controls

- Use the search field (top-right) to filter applications by name.
- Use pagination controls at the bottom of the page to navigate.
- Adjust the number of items per page if needed.
- Use the column selector (if available) to customize visible columns.

## 3.3.2 Application Management > Groups

### What Is an Application Group?

An **application group** is a container used to **link applications with profiles**.

A group defines:

- Which **applications** belong together
- Which **profiles** can access those applications

Application groups serve to:

- Connect *Applications* to *Profiles*
- Organize applications into logical sets

#### Important

- Applications are assigned **to groups**
- Profiles are assigned **to groups**
- Users gain access to applications **through profiles → groups → applications**

### Application Groups Page Overview

The **groups** page displays a list of all configured groups.

From this page, administrators can:

- create new groups
- view group details
- edit group configuration
- delete groups
- search for groups

- navigate using pagination controls

Each row represents one group and includes management actions.

## Managing Application Groups

### Creating a New Group

Click **+ New Group** (top-right corner).

Enter the required information.

Save the group.

After creation, applications and profiles can be assigned to this group.

### Group Configuration (Field-by-Field)

#### 1. Name

The name of the group.

- Should reflect purpose or category
  (e.g., *Engineering Tools*, *Finance Apps*, *Remote Access*)

#### 2. Add Applications

Select one or more applications to include in this group.

- These applications will be available to assigned profiles
- Multiple applications can be added

#### 3. Add Profiles

Select one or more profiles.

- Users assigned to these profiles will gain access to the group’s applications

### How Access Works

Access is resolved as follows:

**User → profile → group → applications**

- A user is assigned to a **profile**
- The Profile is linked to one or more **groups**
- Each Group contains **applications**
- The user can access those applications

### Viewing a Group

Click the **View** icon to see group details.

This allows you to review the configuration without editing.

### Editing a Group

Click the **Edit** icon to modify the group.

You can update:

- Name
- Assigned applications
- Assigned profiles

Changes take effect after saving.

### Deleting a Group

Click the **Delete** icon to permanently remove the group.

⚠ Deleting a group does not delete the applications inside it.
Applications remain in the system but are no longer linked through that group.

### Search and Table Controls

Use the search field (top-right) to filter groups by name.

Use pagination controls at the bottom to navigate between pages.

Adjust the number of items per page if needed.

Use the column selector (if available) to customize visible columns.

## 3.3.3 Application Fields

### Application Configuration Fields

Below is a complete list of all fields available when creating or editing an application in VISULOX.

#### 1. Name

The display name of the application shown to users.
Should clearly identify the resource (e.g., *Firefox*, *Remote Desktop*, *SAP Client*).

#### 2. Comment

Optional internal note for administrators.
Useful for documentation or operational context.

#### 3. Icon Upload

Upload an icon (SVG, PNG, JPG, GIF – max 300KB).
Displayed in the user interface to visually identify the application.

#### 4. Application Command

The executable path or command used to start the application.
Example: `/usr/bin/firefox`

#### 5. Argument for Command

Optional command line arguments passed to the application.
Example: `--kiosk --profile user1`

#### 6. Environment Variables

Key-value pairs passed to the application environment.
Format: `KEY=value`
Example: `DB_HOST=localhost`

#### 7. R (Recording)

Enables session recording.

#### 8. IU (Isolated User Mode)

Runs each session in an isolated environment per user.

#### 9. OCR

Enables text recognition within the session.

#### 10. WM (Window Manager)

Adds a VISULOX managed window manager to the session.

#### 11. No SC (Disable Session Control)

Disables session lifecycle control mechanisms.

#### 12. Sync

Shares the user's Transit Zone across sessions.

#### 13. Upload

Mounts an upload directory for file transfer.

#### 14. SCX

Moves the session control bar to the bottom.

#### 15. D (Startup Delay)

Adds a delay before the application starts.

#### 16. Connection Method

Defines how the application is accessed:

- **SSH** – Linux application via SSH
- **Direct** – Direct/local execution
- **RDP** – Remote Desktop (Windows)

#### 17. Single Sign-On (SSO)

Controls authentication behavior:

- **Disabled** – No SSO
- **Enabled** – Uses stored credentials

- **Auto Provisioning** – Creates and logs in users automatically

#### 18. Number of Sessions

Defines how many concurrent sessions are allowed:

- **Unlimited**
- **Limited**

#### 19. Maintain Connection

Controls session persistence:

- **Enabled** – Session remains active after disconnect
- **Disabled** – Session is terminated on disconnect

#### 20. Application Type

Defines the nature of the application:

- **X11 GUI App** – Graphical Linux app
- **X11 Character App** – Terminal-based app
- **X11 Legacy Character App** – Older terminal apps
- **Not Important** – No specific type
- **Remote Desktop App** – Full desktop session

#### 21. Initial Resolution

Defines the session display resolution:

- SD / HD / FHD / QHD / Auto

#### 22. X11 Server Type

Specifies the backend display server:

- **XDummy** – Lightweight virtual display
- **XVFB** – Virtual framebuffer
- **Auto** – Automatically selected

#### 23. Window Type

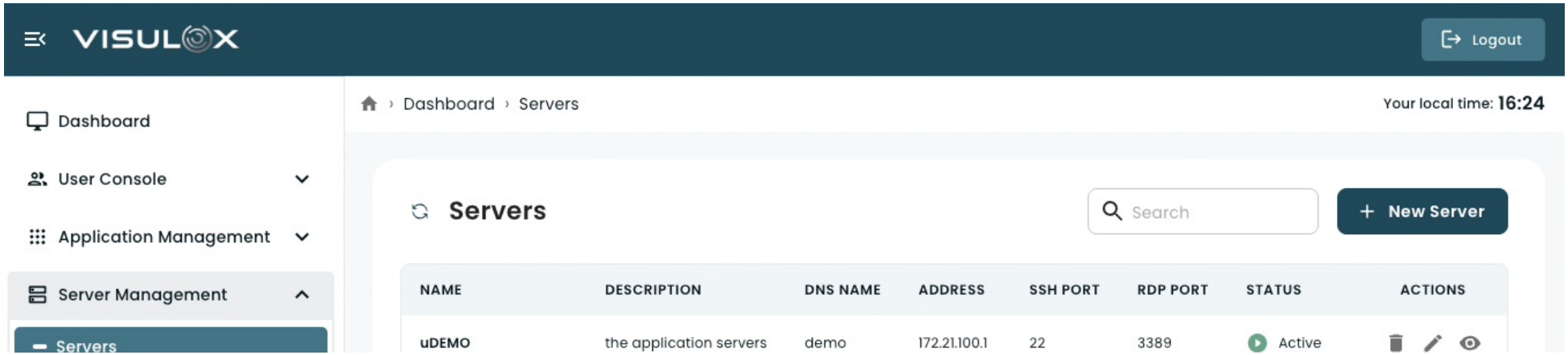
Defines how the session is presented:

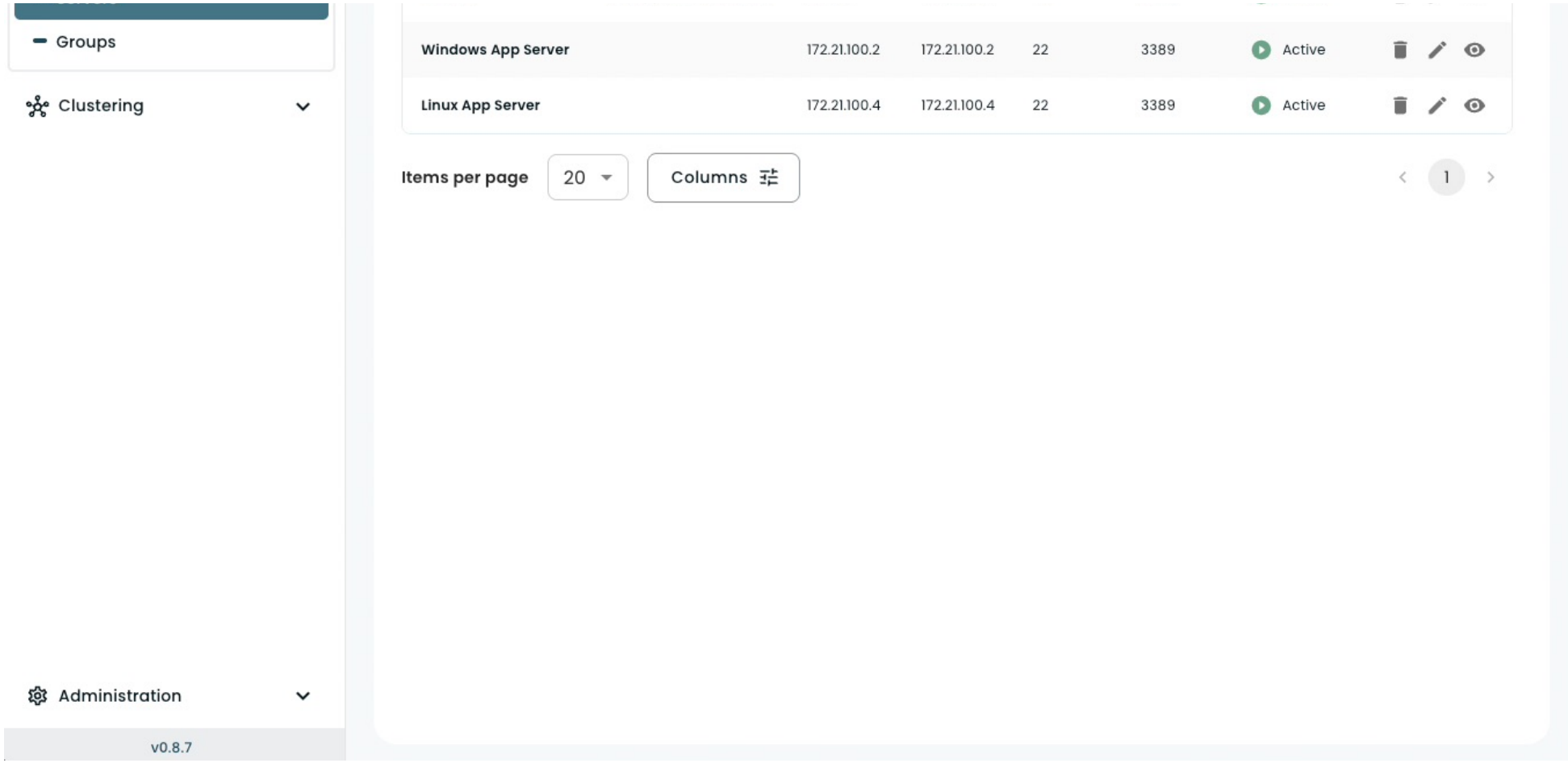
- **Seamless** – Application appears as a native window
- **Desktop** – Full remote desktop

## 3.4 Server Management

- Server Management > Servers
- Server Management > Server Groups

### 3.4.1 Server Management > Servers




Groups
Clustering
Administration
v0.8.7
Windows App Server 172.21.100.2 172.21.100.2 22 3389 Active
Linux App Server 172.21.100.4 172.21.100.4 22 3389 Active
Items per page 20
Columns
1

## What is a Server in VISULOX?

A **Server** represents a system that hosts applications or provides remote resources.

Servers are required so that VISULOX knows:

- where applications are running
- how to connect to them
- which ports are used
- whether the server is reachable

Applications rely on configured servers to function properly.

If a server is not configured or is inactive, applications assigned to it may not work.

## Servers Page Overview

The **servers** page provides a centralized overview of all configured servers.

Each server entry displays important connection and status information, including:

- Server name
- Description
- DNS name
- Address
- SSH port
- RDP port
- Current status (Active / Inactive / Maintenance)

From this page, administrators can:

- add new servers
- review server details
- modify server settings
- remove servers

- search and filter entries
- navigate using pagination

The **status** indicator reflects the availability or intended usage state of the server.

## Managing Servers

### Adding a New Server

Click **+ New Server** (top-right corner).

Enter the required server details.

Save the configuration.

After saving, the server becomes available for application assignments.

### Server Configuration (Field-by-Field)

#### 1. Name

The name of the server

- should follow organizational naming conventions
- is used to identify the server within the system

#### 2. Description

Optional description of the server.

- Can include role, purpose, or configuration details
  (e.g., *Production app server*, *RDP host for finance team*)

#### 3. DNS Name

The DNS name or network identifier of the server

- can be a hostname or IP address
- is used for resolving the server in network communication

#### 4. Address

The fully qualified domain name (FQDN) or address

- Example: `server.domain.com`
- is used for establishing connections

#### 5. SSH Port

Port used for SSH connections.

- Default: `22`
- Required for Linux-based or SSH-based applications

#### 6. RDP Port

Port used for Remote Desktop (RDP) connections.

- Default: `3389`
- Required for Windows-based remote sessions

#### 7. Status

Defines the operational state of the server:

- **Active**
  Server is available for use

- **Inactive**
  Server is disabled and not used
- **Maintenance**
  Server is temporarily unavailable (e.g., updates, troubleshooting)

#### Viewing a Server

Click the **View** icon to open server details in read-only mode.

#### Editing a Server

Click the **Edit** icon to update server configuration such as:

- Name
- Description
- DNS name
- Address
- Ports
- Status

Changes take effect after saving.

#### Deleting a Server

Click the **Delete** icon to permanently remove the server.

⚠ If applications are assigned to this server, they may stop functioning after deletion.

### Search and Table Controls

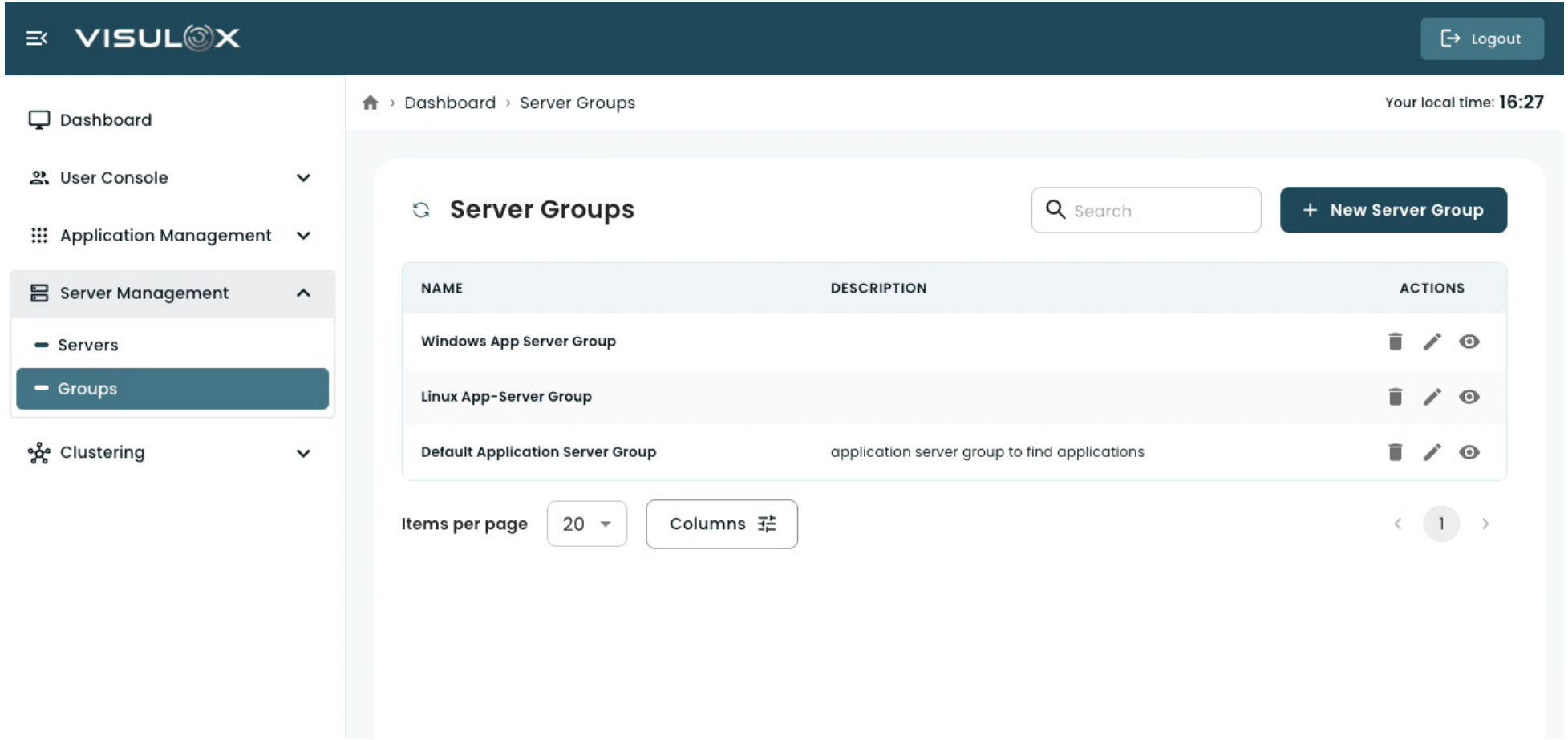
Use the search field (top-right) to filter servers by name.

Use pagination controls at the bottom to navigate between pages.

Adjust the number of items per page as needed.

Use the column selector (if available) to customize visible columns.

### 3.4.2 Server Management > Server Groups




## What is a Server Group?

A **server Group** is a container used to **link servers with applications**.

A server group defines:

- which **servers** are grouped together
- which **applications** can run on those servers

Server Groups help:

- connecting *applications* to *servers*
- organizing infrastructure into logical sets
- enabling applications to run on multiple servers

They do not replace individual server configuration.
Each server must still be defined separately.

## Server Groups Page Overview

The **server groups** page displays all configured server groups.

For each group, you can see:

- Group name
- Description (if defined)

From this page, administrators can:

- create new server groups
- view group details
- modify group information
- delete server groups
- search and filter entries
- navigate using pagination

## Managing Server Groups

### Creating a New Server Group

Click **+ New Server Group** (top-right corner).

Enter the required information.

Save the group.

After creation, servers and applications can be assigned to this group.

### Server Group Configuration (Field-by-Field)

#### 1. Name

The name of the server group

- should reflect purpose or environment
  (e.g., *Production Servers*, *RDP Hosts*, *Linux Cluster*)

#### 2. Description

Optional description of the group

- can include purpose, scope, or configuration details

#### 3. Add Servers

Select one or more servers.

- Defines where the applications will run
- Multiple servers can be included for scalability or redundancy

#### 4. Add Applications

Select one or more applications.

- These applications will be linked to the selected servers
- Enables applications to run on any server within the group

## How It Works

### Application → server group → servers

- An application is linked to a **server group**

- The server group contains one or more **servers**
- The application can run on any server in that group

### Viewing a Server Group

Click the **View** icon to review the group details in read-only mode.

### Editing a Server Group

Click the **Edit** icon to update:

- Name
- Description
- Assigned servers
- Assigned applications

Changes take effect after saving.

### Deleting a Server Group

Click the **Delete** icon to permanently remove the group.

> ⚠ Deleting a server group does not delete the servers or applications inside it.
> They remain configured but are no longer linked through that group.

### Search and Table Controls

Use the search field (top-right) to filter groups by name.

Use pagination controls at the bottom to navigate between pages.

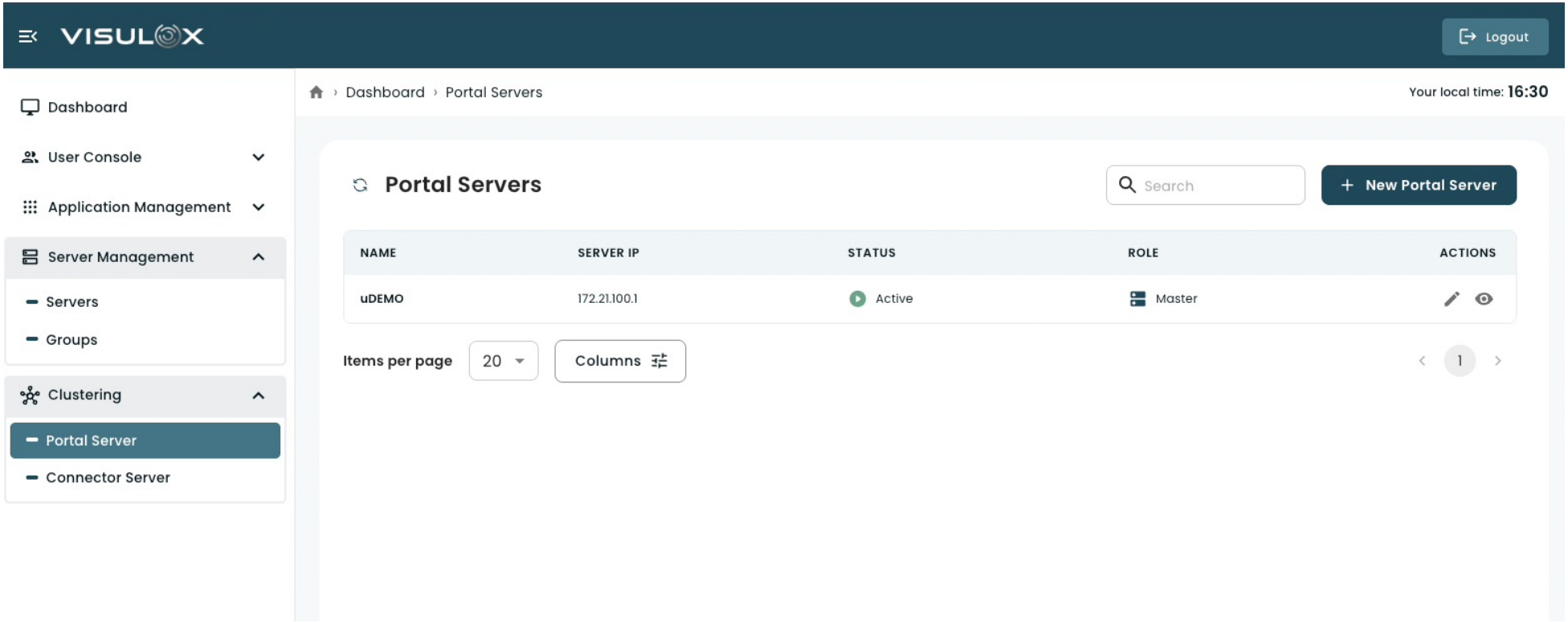
Adjust the number of items per page if needed.

Use the column selector (if available) to customize visible columns.

## 3.5 VISULOX Cluster Management

- Clustering > Portal Servers
- Clustering > Connector Servers

### 3.5.1 Clustering > Portal Servers




## What is a Portal Server?

A **Portal Server** is a VISULOX instance responsible for:

- handling user authentication
- managing sessions
- serving the administration interface
- coordinating application access

In clustered environments, multiple Portal Servers can work together to provide:

- high availability
- load distribution
- redundancy

Each Portal Server has a defined role within the cluster.

## Roles

### Master

The **Master Portal Server** is the primary node in the cluster.

It is responsible for:

- Cluster coordination
- configuration management
- synchronization with other nodes

Only one Master server exists within a cluster.

### Worker

A **Worker Portal Server** operates as a secondary node and:

- handles user sessions
- executes workload assigned by the Master
- contributes to load distribution and scalability

## Portal Servers Page Overview

The **Portal Servers** page provides an overview of all configured Portal nodes.

For each server, the system displays:

- Server name
- Server IP address
- External DNS name
- Current status (Active / Inactive)
- Cluster role (Master / Worker)

From this page, administrators can:

- add a new Portal Server
- review server details
- modify configuration
- monitor cluster status

## Managing Portal Servers

### Adding a New Portal Server

Click **+ New Portal Server**.

Enter the required server details.

Save the configuration.

After configuration, the server can join the cluster.

### Portal Server Configuration (Field-by-Field)

#### 1. Name

The name of the Portal Server

- should follow organizational naming conventions
- is used to identify the node within the cluster

#### 2. Server IP

The IP address assigned to the Portal Server

- is used for internal communication
- must be reachable from other nodes

#### 3. External DNS Name

The public DNS name of the Portal Server

- Example: `portal.company.com`
- is used by users to access the system

#### 4. Session Peer DNS Name

The DNS name used for session peer connectivity

- is used for communication between cluster nodes
- hhould resolve correctly within the internal network

#### 5. Role

Defines the role of the server within the cluster:

- **Master**
  Primary node responsible for coordination
- **Worker**
  Secondary node handling workload and sessions

## Viewing a Portal Server

Click the **View** icon to review the server details in read-only mode.

## Editing a Portal Server

Click the **Edit** icon to update configuration settings such as:

- Name
- IP address

- DNS settings
- Role

Changes take effect after saving.

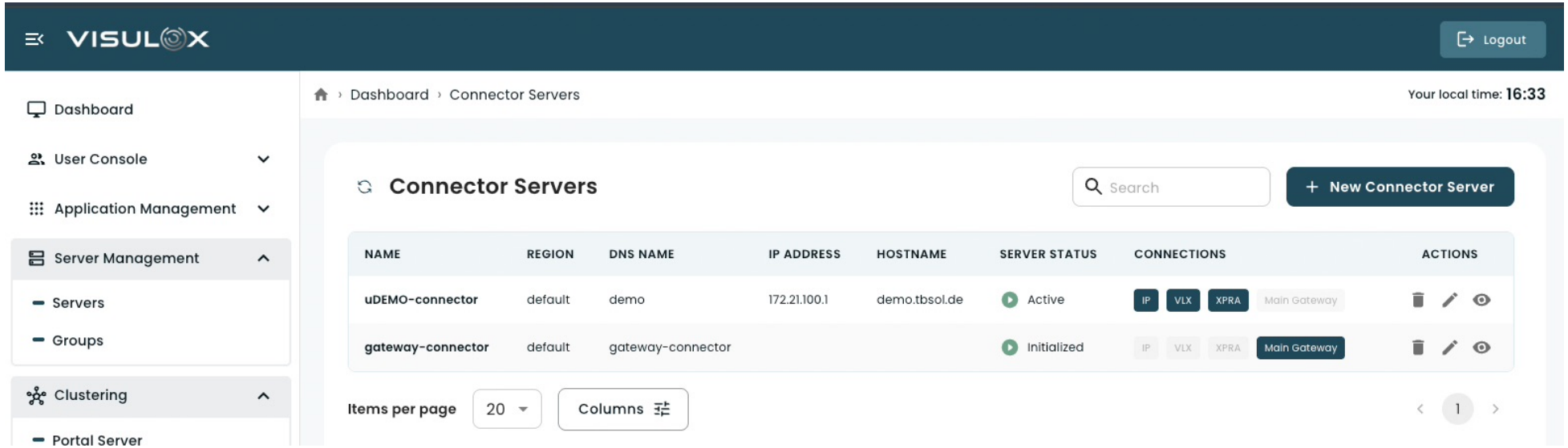
### Status Indicator

The Status column shows whether the Portal Server is:

- **Active** → Running and reachable
- **Inactive** → Not reachable or not running

Monitoring this status helps ensure cluster stability.

## 3.5.2 Clustering > Connector Servers




## What is a Connector Server?

A **Connector Server** is responsible for handling remote session traffic between users and backend application servers.

While the Portal Server manages authentication and coordination, the Connector Server:

- establishes session connections
- routes traffic to target servers
- manages protocol-level communication
- acts as a gateway between users and application infrastructure

Connector Servers are essential in distributed or clustered deployments.

## Key Concepts

### Region

Connector Servers can be organized by region.

- Useful for geographically distributed environments
- Helps control routing and locality of connections

### Server Status

Indicates the operational state of the Connector:

- **Active** → Fully operational
- **Initialized** → Registered but not fully active

Monitoring status ensures proper routing and availability.

### Connections

Defines which connection types are enabled on the Connector Server:

- **XPRA** → For graphical Linux sessions
- **VLX** → VISULOX protocol
- **Main Gateway** → Central routing gateway

These determine how the Connector handles traffic.

## Connector Servers Page Overview

The page displays all configured Connector nodes with:

- Name
- Region

- DNS name
- IP address
- Hostname
- Server status
- Active connection types

From this page, administrators can:

- add new Connector Servers
- review configuration details
- modify existing connectors
- remove connectors
- search and filter entries
- navigate using pagination

## Managing Connector Servers

### Adding a New Connector Server

Click **+ New Connector Server**.

Enter the required configuration details.

Save the configuration.

After setup, the Connector Server can join and operate within the system.

### Connector Server Configuration (Field-by-Field)

#### General Information

##### 1. Name

The name of the Connector Server

- is used for identification within the system
- should follow naming conventions

#### 2. Description

Optional description

- can include purpose, environment, or role

### Addressing & Host Configuration

#### 3. Region

Defines the region of the Connector Server.

- Used for grouping and routing decisions

#### 4. DNS Name

The DNS name of the Connector Server

- is used for network resolution

#### 5. IP Address

The IP address of the Connector Server

- must be reachable by other system components

#### 6. Hostname

The hostname of the Connector Server

- typically matches system hostname

#### Connection & Routing

##### 7. Can connect with XPRA

Enables handling of XPRA-based sessions.

##### 8. Can connect with VLX

Enables VISULOX protocol connections.

##### 9. Can connect with Main Gateway

Allows routing through the main gateway.

#### Access Control

##### 10. Available on IP Address

Restricts availability based on IP address.

- When enabled, the connector is accessible via its IP

## Viewing a Connector Server

Click the **View** icon to review the configuration in read-only mode.

## Editing a Connector Server

Click the **Edit** icon to update:

- General information
- Addressing details

- Connection settings

Changes take effect after saving.

### Deleting a Connector Server

Click the **Delete** icon to permanently remove the Connector.

⚠ Removing a Connector may interrupt active or future session routing.

## 3.6 Access Tokens for external access to the VISULOX Dashoard

### 3.6.1 General

The possibility to create Access Tokens and configure the permissions for the API access is implemented in the VISULOX Dashboard.

With these tokens a user with administrative rights is able to:

- create access
- revoke access
- set detailed permissions

### 3.6.2 Prerequisites

For this test setup one user with administrative access to the VISULOX PORTAL is needed.

Dashboard URL: https://<*hostname*.domain>:<*port*>/dashboard/login

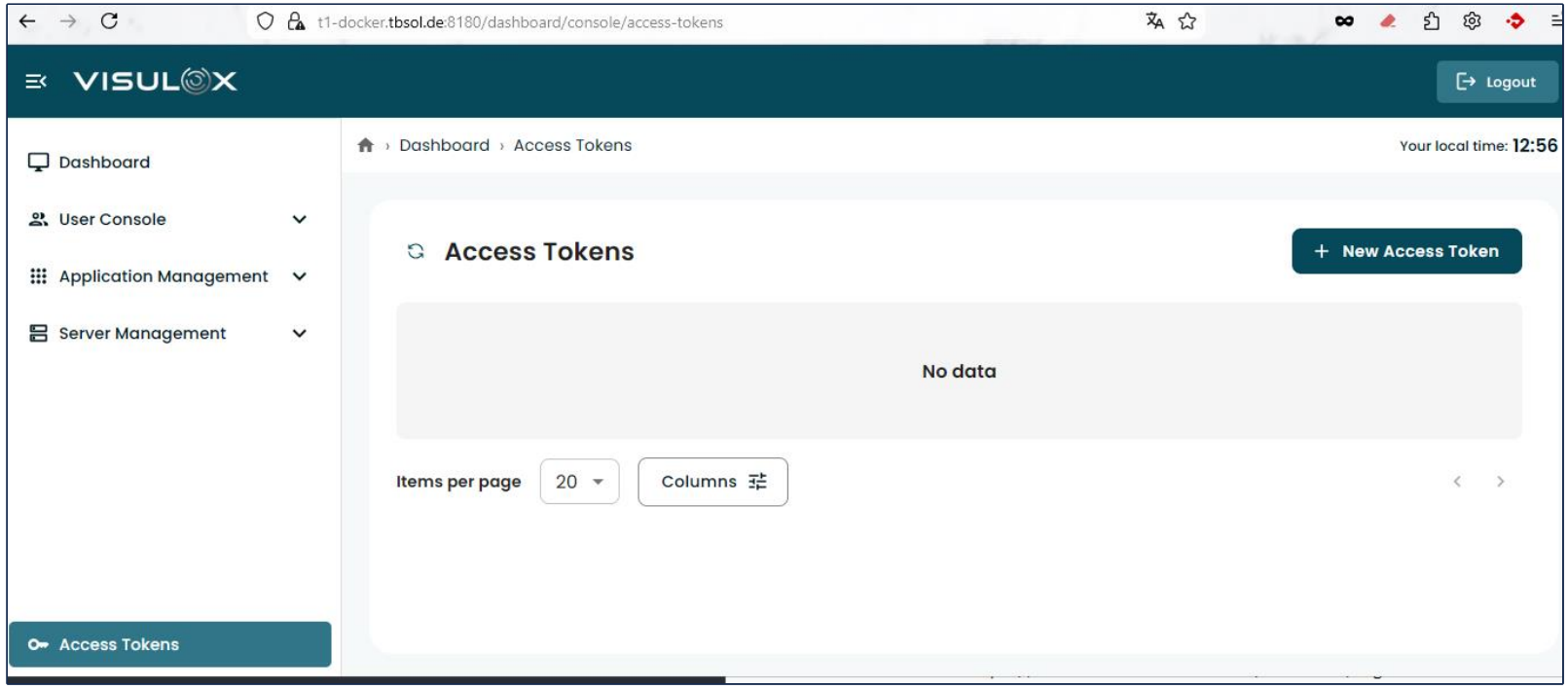
Example user:

| Name | Profile | Login name |
| --- | --- | --- |
| Administrator | Administrator User Profile | develadmin |

### Enable access via tokens for API access in the VISULOX Dashboard

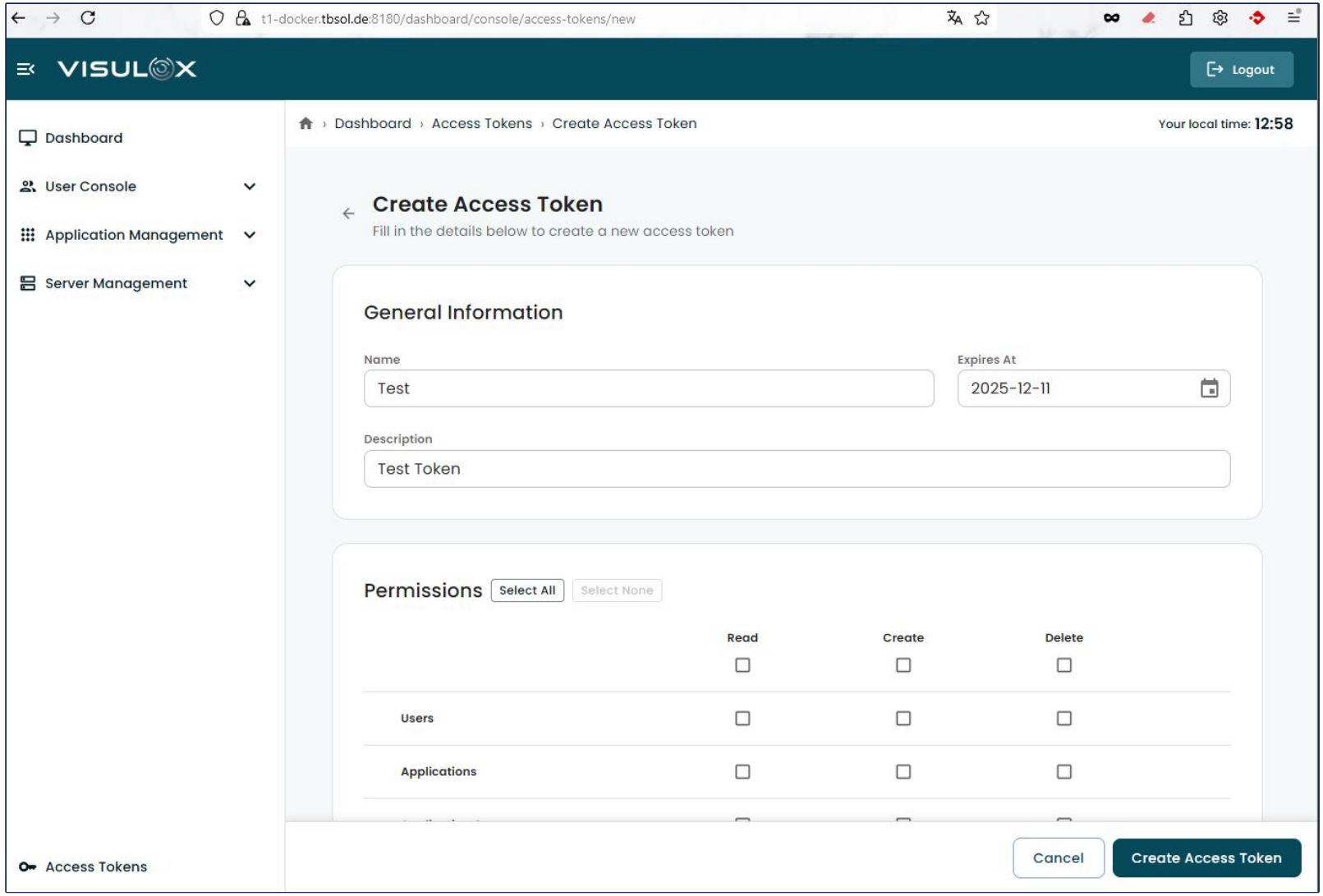
Administrator logs into the VISULOX Dashboard.

Once logged in, the administrator selects **Access Tokens** on the bottom left:



The administrator creates a new access token via the **+ New Access Token** button.

The general information fields have to be filled out:

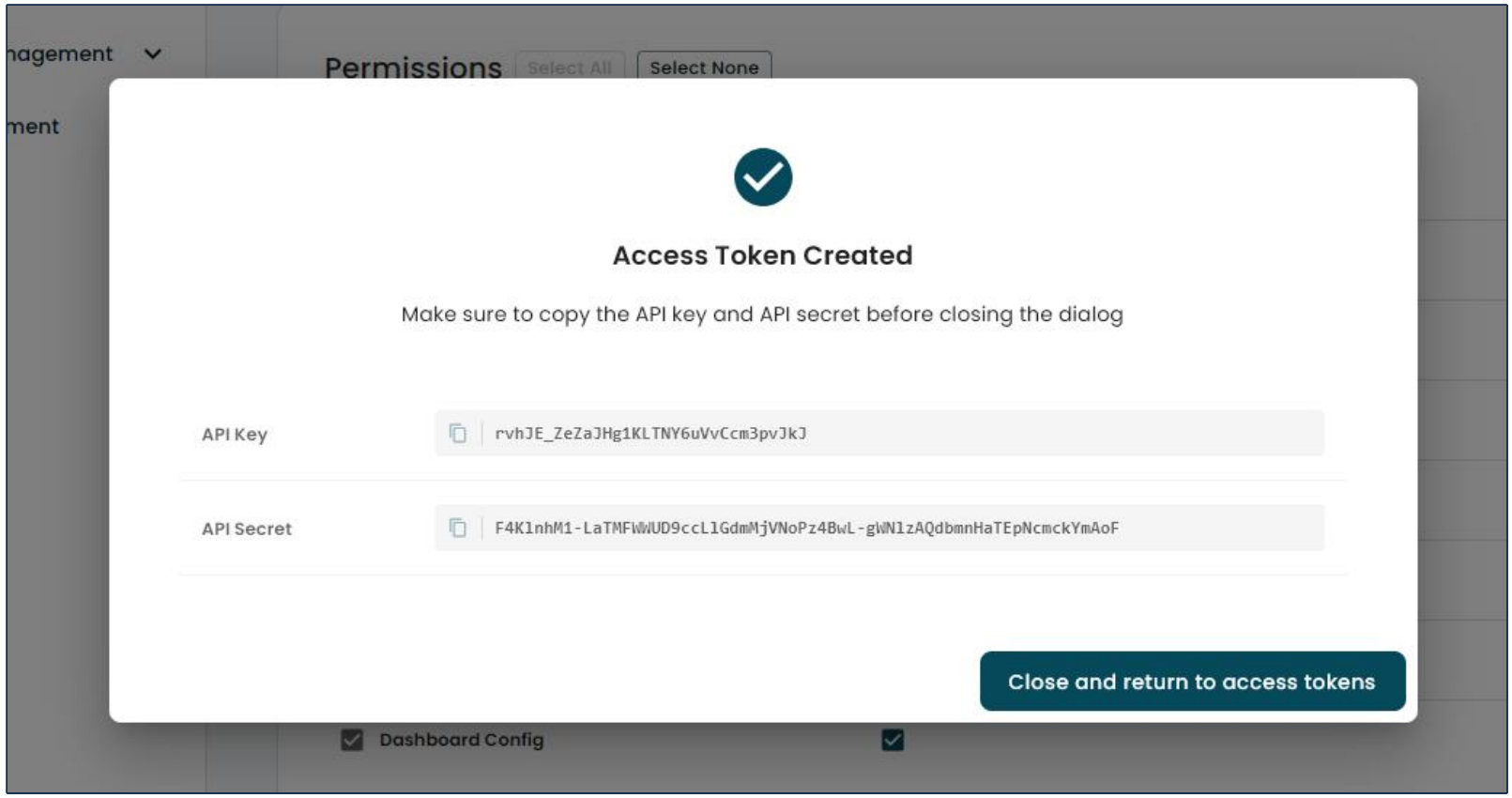

t1-docker.tbsol.de:8180/dashboard/console/access-tokens/new
VISULOX
Logout
Dashboard
User Console
Application Management
Server Management
Access Tokens
Dashboard › Access Tokens › Create Access Token
Your local time: 12:58
Create Access Token
Fill in the details below to create a new access token
General Information
Name
Test
Expires At
2025-12-11
Description
Test Token
Permissions
Select All
Select None
Read
Create
Delete
Users
Applications
Cancel
Create Access Token

Name (required) / Expiration date (required) / Description (optional) / detailed permissions (read, write, delete).

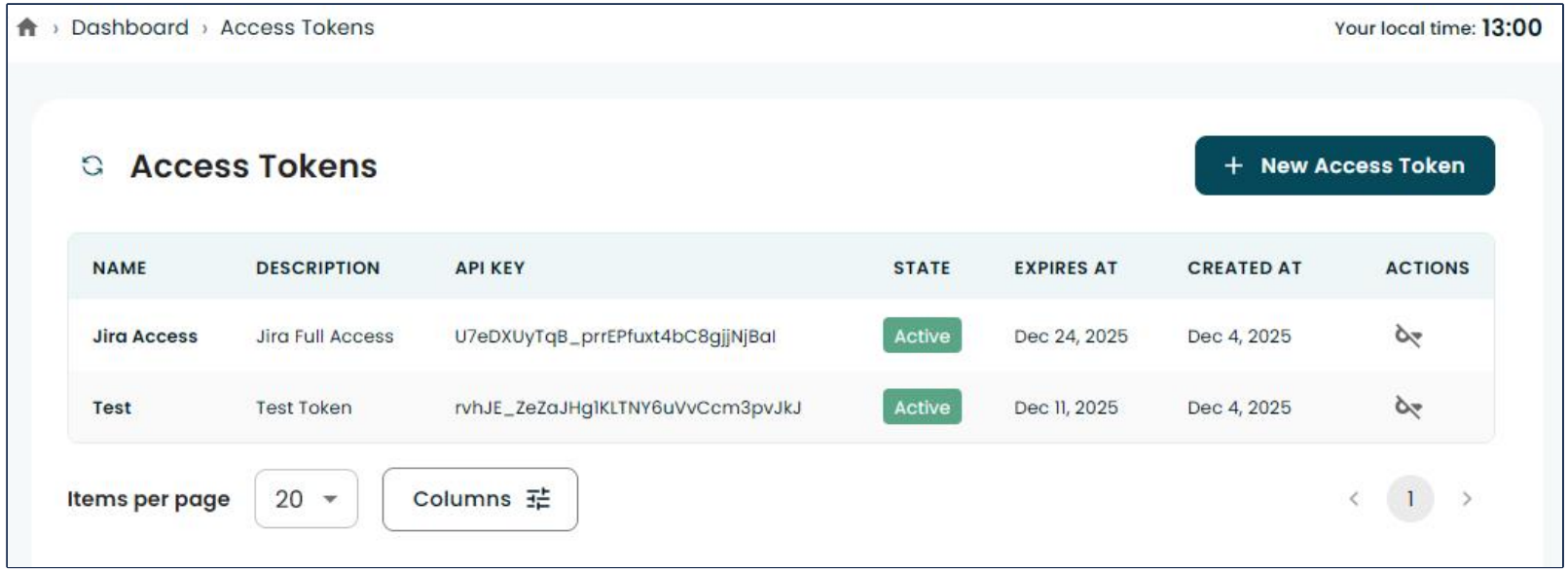
With the **Create Access Token** button, the token will be created.

A new window is opened and the following items are copied into clipboard on click:

- API key
- API Secret




When done, using the **Close and return to access tokens** button, the Access token is shown in token list:




The Access Token can be revoked with the Revoke button in the table (Actions).

# 4 Monitoring & Log Inspection Guide

*Purpose:* *How to monitor VISULOX services, inspect logs, change log level, and validate runtime health on Linux systems.*

*Scope:* *VISULOX Portal, VISULOX Connector, VISULOX Core.*


- 1. Installed Version Verification
- 2. VISULOX Core
- 3. VISULOX Portal
- 4. VISULOX Connector
- 5. Process Verification
- 6. Port Verification
- 7. Typical Operational Flow after Upgrade
- 8. Common Failure Indicators & Immediate Actions
- 9. Permissions and Safe Editing
- 10. Check Session Startup Logs


## 4.1 1. Installed Version Verification

Verify installed RPM packages:

```
rpm -qa | grep -E "visulox|vlx"
```

## 4.2 2. VISULOX Core

Service Status:

```
systemctl status visulox
# Healthy state indicator: Active: active (running)

 visulox integrity
```

```
 # To check component status
```

Logs:

```
journalctl -u vlx-core.service -f
journalctl -u vlx-core.service -n 200
```

Restart:

```
systemctl restart visulox
```

Core Configuration Validation:

```
visulox config list

# Exmaples:
visulox config -name role
visulox config -name visulox-portal.hostname

# Edit example:
visulox config edit -name visulox-portal.hostname='raw.tbsol.de'
```

## 4.3 3. VISULOX Portal

### 4.3.1 3.1. Service Status and Logs

Status:

```
systemctl status vlx-portal
```

```
# Healthy state indicator: Active: active (running)
```

Logs:

```
journalctl -u vlx-portal -f
```

File-Based Logs (Default path):

```
/var/log/visulox-portal/portal.log

# Monitor:
tail -f /var/log/visulox-portal/portal.log
```

## 4.3.2 3.2. Set log level and log file path

### 3.2.1. Portal

Edit configurations:

```
sudo vi /opt/visulox-portal/configs/config.json
```

Update `Logger` block:

```
"Logger": {
  "Level": "debug",
  "ExtraFilePath": "/var/log/visulox-portal/portal.log"
}
```

3. Ensure file permissions:

```
sudo mkdir -p /var/log/visulox-portal
sudo touch /var/log/visulox-portal/portal.log
sudo chown vlx:vlxgroup /var/log/visulox-portal/portal.log
sudo chmod 664 /var/log/visulox-portal/portal.log
```

4. Restart:

```
sudo systemctl restart vlx-portal
```

⚠ Use `debug` only temporarily.

### 3.2.2. Connector

Edit configurations:

```
sudo vi /opt/visulox-portal-connector/configs/config.json
```

Update `Logger` block:

```
"Logger": {
  "Level": "debug",
  "ExtraFilePath": "/var/log/visulox-portal-connector/connector.log"
}
```

3. Ensure file permissions:

```
sudo mkdir -p /var/log/visulox-portal-connector
sudo touch /var/log/visulox-portal-connector/connector.log
sudo chown vlx:vlxgroup /var/log/visulox-portal-connector/connector.log
sudo chmod 664 /var/log/visulox-portal-connector/connector.log
```

4. Restart:

```
sudo systemctl restart vlx-connector
```

⚠ Use `debug` only temporarily.

### 4.3.3 3.5 UI Validation (Hostname Required)

VISULOX may not work correctly via IP.
Access must use the configured hostname.

Example:

```
https://raw.tbsol.de:8180/ui4/login
```

Do not use:

```
https://<SERVER_IP>:8180
```

## 4.4 4. VISULOX Connector

### 4.4.1 4.1 Service Status

```
systemctl status vlx-connector
```

### 4.4.2 4.2 Real-Time Connector Logs (systemd)

```
journalctl -u vlx-connector -f
```

### 4.4.3 4.3 File-Based Connector Logs (if enabled)

Default path:

```
/var/log/visulox-portal-connector/connector.log
```

Monitor:

```
tail -f /var/log/visulox-portal-connector/connector.log
```

### 4.4.4 4.4 Set Connector Log Level

1. Edit:

```
sudo vi /opt/visulox-portal-connector/configs/config.json
```

2. Update:

```
"Logger": {
  "Level": "debug",
  "ExtraFilePath": "/var/log/visulox-portal-connector/connector.log"
}
```

3. Ensure permissions:

```
sudo mkdir -p /var/log/visulox-portal-connector
sudo touch /var/log/visulox-portal-connector/connector.log
sudo chown vlx:vlxgroup /var/log/visulox-portal-connector/connector.log
sudo chmod 664 /var/log/visulox-portal-connector/connector.log
```

4. Restart:

```
sudo systemctl restart vlx-connector
```

## 4.5 5. Process Verification

```
ps aux | grep -E "visulox|vlx" | grep -v grep
```

Expected:

- visulox
- vlx-portal
- vlx-connector

## 4.6 6. Port Verification

Check listening ports:

```
ss -tulpen | grep -E "8180|vlx|visulox|connector|portal"
```

Validate:

- Expected ports listening
- Bound to correct interface

- No port conflicts

## 4.7 7. Typical Operational Flow after Upgrade

1. Verify RPM versions
2. Check service status
3. Restart services (if required)
4. Monitor logs in real time
5. Validate hostname-based UI access
6. Confirm no restart loops

## 4.8 8. Common Failure Indicators & Immediate Actions

### 4.8.1 8.1 Restart Loop

```
journalctl -u <service> -n 200
```

### 4.8.2 8.2 Configuration Error

Inspect:

```
journalctl -u <service>
visulox config list
```

### 4.8.3 8.3 Port Conflict

```
ss -tulpen | grep <port>
```

## 4.9 9. Permissions and Safe Editing

Ensure log files have correct ownership.

Portal:

```
chown vlx:vlxgroup /var/log/visulox-portal/portal.log
```

Connector:

```
chown vlx:vlxgroup /var/log/visulox-portal-connector/connector.log
```

Restart service after changes.

## 4.10 10. Check Session Startup Logs

For each application session, a directory is created under:

```
/var/visulox-portal-connector/xpra
```

Run the following command to list the session directories sorted by date and time:

```
ls -alihtr
```

The **latest directory corresponds to the most recent session**.

To find the correct folder, note that in **Web Mode** each session has a unique identifier (**UUID**) visible in the URL.
This UUID represents the **EmulatorSession ID**, and the corresponding directory in this path is created with the **same name**.

Inside the session directory, you can find the session startup logs in:

```
app_launcher.log
```

# 5 Database Backup Guide

| | |
|---|---|
| **Scope:**<br>Automatic and manual PostgreSQL database backup and restore for VISULOX. | • Purpose<br>• Automatic Backup<br>• Backup Retention Policy<br>• Backup Location<br>• Manual Backup<br>  • Step 1 - Run backup inside backup container<br>  • Step 2 - Verify backup was created<br>• Manual Restore<br>  • Step 1 - Stop Portal and Connector services<br>  • Step 2 - Restore database from backup<br>  • Step 3 - Restart services<br>  • Step 4 - Verify system status<br>• Verify Backup System<br>• Troubleshooting |

## 5.1 Purpose

This document describes how VISULOX database backups work, how to create manual backups, and how to restore the database if needed.

VISULOX automatically creates periodic backups of its PostgreSQL database after setup is completed using `setup-master`.

This ensures that:

- database contents can be restored in case of failure
- historical backups are retained
- backup and retention are fully automated
- manual backup and restore are available when needed

## 5.2 Automatic Backup

VISULOX performs a full PostgreSQL database dump:

- Every day at **08:34 AM**
- Backup files are stored in:

```
/var/visulox-portal/dbs/backups
```

Each backup contains a complete snapshot of the database.

## 5.3 Backup Retention Policy

VISULOX automatically removes old backups.

Retention policy:

- Backups older than **30 days** are deleted automatically

## 5.4 Backup Location

Backup directory:

```
/var/visulox-portal/dbs/backups
```

List backups:

```
ls -lh /var/visulox-portal/dbs/backups
```

Example:

```
-rw-r--r-- 1 vlx vlxgroup 45M Feb 12 08:34 visulox_backup_2026-02-12.sql
-rw-r--r-- 1 vlx vlxgroup 46M Feb 13 08:34 visulox_backup_2026-02-13.sql
```

## 5.5 Manual Backup

Manual backups can be created at any time.

### 5.5.1 Step 1 - Run backup inside backup container

```
sudo vlx-docker exec vlx-db-backup /usr/local/bin/backup.sh
```

This will create a new backup file in:

```
/var/visulox-portal/dbs/backups
```

### 5.5.2 Step 2 - Verify backup was created

```
ls -lh /var/visulox-portal/dbs/backups
```

You should see a new backup file with the current date and time.

## 5.6 Manual Restore

Use this procedure to restore the database from a backup file.

⚠ **WARNING:** This will overwrite the current database.

### 5.6.1 Step 1 - Stop Portal and Connector services

```
sudo systemctl stop vlx-portal
sudo systemctl stop vlx-connector
```

### 5.6.2 Step 2 - Restore database from backup

Run restore inside the PostgreSQL container:

```
sudo vlx-docker exec -i vlx-postgres psql -U postgres -d visulox < /var/visulox-portal/dbs/backups/<backup-file>.sql
```

Example:

```
sudo vlx-docker exec -i vlx-postgres psql -U postgres -d visulox < /var/visulox-portal/dbs/backups/visulox_backup_2026-02-13.sql
```

### 5.6.3 Step 3 - Restart services

```
sudo systemctl start vlx-portal
sudo systemctl start vlx-connector
```

### 5.6.4 Step 4 - Verify system status

Check portal service:

```
sudo systemctl status vlx-portal
```

Check connector service:

```
sudo systemctl status vlx-connector
```

Check logs:

```
tail -f /var/log/visulox-portal/portal.log
```

## 5.7 Verify Backup System

Check backup container:

```
sudo vlx-docker ps | grep vlx-db-backup
```

Check backup schedule:

```
sudo vlx-docker exec vlx-db-backup crontab -l
```

Expected output:

```
34 8 * * * /usr/local/bin/backup.sh
```

## 5.8 Troubleshooting

Check backup container logs:

```
sudo vlx-docker logs vlx-db-backup
```

Check PostgreSQL container:

```
sudo vlx-docker ps | grep postgres
```

Check backup directory:

```
ls -ld /var/visulox-portal/dbs/backups
```

Expected owner:

```
vlx:vlxgroup
```

# 6 Health Check for VISULOX Portal

## 6.1 Purpose

The `health-check` command helps a system administrator verify that a VISULOX Portal installation is usable after installation, upgrade, or maintenance.

The command checks the local machine, required software, host permissions, important filesystem paths, PostgreSQL, KeyDB, REST and SOAP APIs, registered portal and connector servers, and configured LDAP or Active Directory servers.

Use this command when

- a new portal server has been installed.
- a portal server has been upgraded.
- a node was joined to a cluster.
- the portal service is running but users report login, API, session, connector, or directory problems.
- you need a JSON health report for support or automation.

## 6.2 Command

```
/opt/visulox-portal/bin/vlx-portald health-check [flags]
```

If the configuration is stored in a file, pass the global config flag:

```
/opt/visulox-portal/bin/vlx-portald --configFile /opt/visulox-portal/configs/config.json health-check
```

Short form:

```
/opt/visulox-portal/bin/vlx-portald -c /opt/visulox-portal/configs/config.json health-check
```

## 6.3 Permissions

In `host` mode, run the command as `root` or as a user with `sudo` access. The command validates host-level installation paths and service files, so a normal user may not be able to inspect every required item.

## 6.4 Output Modes

### 6.4.1 CLI Report

The default mode prints a formatted report:

```
/opt/visulox-portal/bin/vlx-portald health-check
```

The report includes the VISULOX logo, version information, detailed tables for every check, and a final `Summary` section. The summary is printed at the end so it remains visible after the long report finishes in the terminal.

### 6.4.2 JSON to stdout

Use `--json` or `-j` when another tool needs structured output:

```
/opt/visulox-portal/bin/vlx-portald health-check --json
```

Short form:

```
/opt/visulox-portal/bin/vlx-portald health-check -j
```

### 6.4.3 Save JSON to a File

Use `--output` to write the structured report to a file:

```
/opt/visulox-portal/bin/vlx-portald health-check --output /tmp/visulox-health-check.json
```

This still prints the normal CLI report unless `--no-print` is also used.

### 6.4.4 Save JSON without Printing

Use `--no-print` with `--output` for automation:

```
/opt/visulox-portal/bin/vlx-portald health-check --output /tmp/visulox-health-check.json --no-print
```

`--no-print` is only valid when `--output` is provided.

## 6.5 What the Command Checks

### 6.5.1 Machine

The command reports:

- VISULOX application name
- Hostname
- First non-loopback IPv4 address
- Application environment
- CPU usage
- RAM usage

- Root disk usage

### 6.5.2 Software

The command checks the required software list:

- `vlx-python` , `python` , or `python3`
- `vlx-docker` or `docker` in host mode
- `xpra`
- `visulox` , `vlx-portal` , or `vlx-portald`
- `nginx`
- `python3-paramiko`
- `python3.12-tkinter` , `python3-tkinter` , or `python3-tk`
- `freerdp` , `freerdp2` , `freerdp3` , `xfreerdp` , or `wlfreerdp`
- `jq`
- `unzip`

In container mode, the Docker check is skipped because Docker is required on the host, not inside the container.

### 6.5.3 Host Access

In host mode, the command verifies that the current process is running as root or can run `sudo -n true` . In container mode, this check is treated as available.

### 6.5.4 Filesystem

The command checks important VISULOX installation paths. In host mode, this includes paths such as:

- `/opt/visulox-portal`
- `/opt/visulox-portal/bin/vlx-portald`
- `/opt/visulox-portal/configs`
- `/opt/visulox-portal/tls`
- `/opt/visulox-portal/packages`
- `/var/log/visulox-portal`
- `/etc/systemd/system/vlx-portal.service`
- `/etc/pam.d/vlx-portal`

In container mode, it checks the container-specific portal paths such as `/opt/visulox-portal/bin/vlx-portal` and `/opt/visulox-portal/docker-entrypoint.sh`.

For each path, the report shows whether it exists, whether it is a file or directory, and whether it is readable, writable, or executable by the current process.

### 6.5.5 PostgreSQL

The command checks PostgreSQL by:

1. Opening a write connection to the master database.
2. Opening a read connection to the slave database, or the master database if no slave is configured.
3. Running `SELECT 1`.
4. Creating a temporary table inside a transaction.
5. Inserting and reading a probe row.

The write probe uses a transaction and is rolled back.

### 6.5.6 KeyDB

The command checks KeyDB by:

1. Creating a KeyDB client from configuration.
2. Writing a temporary health-check key.
3. Reading the key back.
4. Deleting the key.

If KeyDB is disabled in configuration, the KeyDB section is reported as failed with a configuration message.

### 6.5.7 REST API

The command sends a local HTTP request to:

```
/api/v1/version
```

The target port and TLS setting come from the Echo Gateway configuration.

### 6.5.8 SOAP API

The command sends a local SOAP keep-alive request to:

```
/soap/services/document/utility
```

The target port and TLS setting come from the SOAP gateway configuration.

### 6.5.9 Portal and Connector Servers

The command reads registered servers from PostgreSQL:

- `portal_server`
- `connector_server`

For each server it prints the database status, last ping time, ping age, and interpreted live status.

Live status is interpreted as:

| Condition | Result |
|---|---|
| Database status is not `active` | `offline` |
| Last ping is empty | `offline` |
| Last ping is less than 1 minute old | `healthy` |
| Last ping is up to 4 minutes old | `warning` |
| Last ping is older than 4 minutes | `dangerous` |

### 6.5.10 LDAP and Active Directory

The command loads LDAP and Active Directory configuration from dynamic configuration when PostgreSQL is available. If dynamic configuration cannot be loaded, it falls back to static authorization configuration.

For each enabled LDAP or Active Directory server, the command opens an LDAP or LDAPS connection, optionally binds with the configured admin user, and optionally runs a base DN probe.

## 6.6 Reading the Summary

The `Summary` section is the fastest way to decide what to investigate first.

| Status | Meaning |
|---|---|
| `OK` | The section passed. |
| `WARNING` | The section is usable but has stale or non-critical health signals. |
| `FAILED` | The section has a failed required check. |
| `SKIPPED` | The section was not applicable or had no configured targets. |

When a section is `FAILED`, inspect the corresponding detailed table above the summary.

## 6.7 Common Problems

### 6.7.1 Missing Software

If the software summary is `FAILED`, install the missing package shown in the `Software` table, then run the command again.

### 6.7.2 Host Access Failed

Run the command as root or configure sudo for the administrator account:

```
sudo -n true
```

The command expects this to succeed in host mode.

### 6.7.3 PostgreSQL Failed

Check the database host, port, database name, username, password, and TLS settings in the VISULOX configuration. Also verify that the server can reach PostgreSQL over the network.

### 6.7.4 KeyDB Failed

Check whether KeyDB is enabled and reachable from the portal server. Verify the KeyDB host, port, password, TLS, and timeout configuration.

### 6.7.5 API Checks Failed

Check whether the REST and SOAP services are running locally and listening on the configured ports. If TLS is enabled, the command uses `<https://127.0.0.1:<port>>`.

### 6.7.6 Server Status is Offline or Dangerous

Check the related portal or connector service. A missing or stale `last_ping_time` usually means the server is not running, cannot reach PostgreSQL, or cannot update its heartbeat.

### 6.7.7 LDAP or Active Directory Failed

Check host, port, TLS mode, certificate behavior, bind user, bind password, and base DN. If the server is disabled in configuration, the command reports it as disabled and does not attempt a connection.