Sentry WebClient

Installation Guide

Version 2026R1

09.03.2026

Copyright, trademarks, and disclaimer

Trademarks

Sentry is a trademark or registered trademarks of IntelexVision. All other trademarks mentioned in this guide are the property of their respective holders. This product may make use of third-party software for which specific terms and conditions may apply.

All names of people and organizations used in the examples in this text are fictitious. Any resemblance to any actual organization or person, living or dead, is purely coincidental and unintended.

Disclaimer

This document is intended for general Information purposes only and due care has been taken in its preparation. Any risk arising from the use of this information rests with the recipient, and nothing herein should be construed as constituting any kind of warranty.

IntelexVision may make changes without prior notice.

This product may make use of third-party software for which specific terms and conditions may apply.

Introduction

Scope

This document provides step-by-step instructions for installing **Sentry WebClient**. It includes all the necessary details to guide users through the installation process.

The scope of this guide is limited to the installation phase and does not include advanced configurations or integration with external systems.

Audience

The following roles are the intended audience for this guide:

- **System Administrators:** Responsible for managing the infrastructure and ensuring the software is installed correctly within the organization's IT environment.
- **Operations Personnel:** Tasked with the daily operation and monitoring of the system to maintain optimal performance and address any operational issues as they arise.
- **IT Support Engineers:** Tasked with resolving technical issues during installation and initial setup.
- **Solution Architects:** Overseeing the installation to ensure alignment with the organization's architecture and project requirements.
- **Technical Implementation Specialists:** Handling the detailed implementation of the software for specific use cases.
- **Product Specialists:** Verifying the successful setup and ensuring that the installed component meets the business needs.

Pre-requisites

Technical Knowledge

- **Basic Operating System Proficiency.** Ability to navigate Windows applications, log into systems, and manage basic settings.
- **Data Entry Skills.** Proficiency in entering data into various Windows-based software applications.
- **System Navigation.** Understanding of how to access and use installed systems, including logging in and performing routine operations.

System Access

- **Administrator Privileges.** Access to a user account with administrator rights on the target Windows machine.
- **Network Permissions.** Make sure you have the required permissions to install software and adjust network settings, such as setting up firewall exceptions.

Preparatory Tasks

- **Review Documentation.** Familiarize yourself with this guide and any other documentation.
- **Backup Policies.** Make sure there's a backup plan ready for any existing software or data that could be impacted during installation.
- **Dependencies Check.** Verify that all required software components or services (e.g., runtime environments, frameworks) are pre-installed or accessible.

System Overview

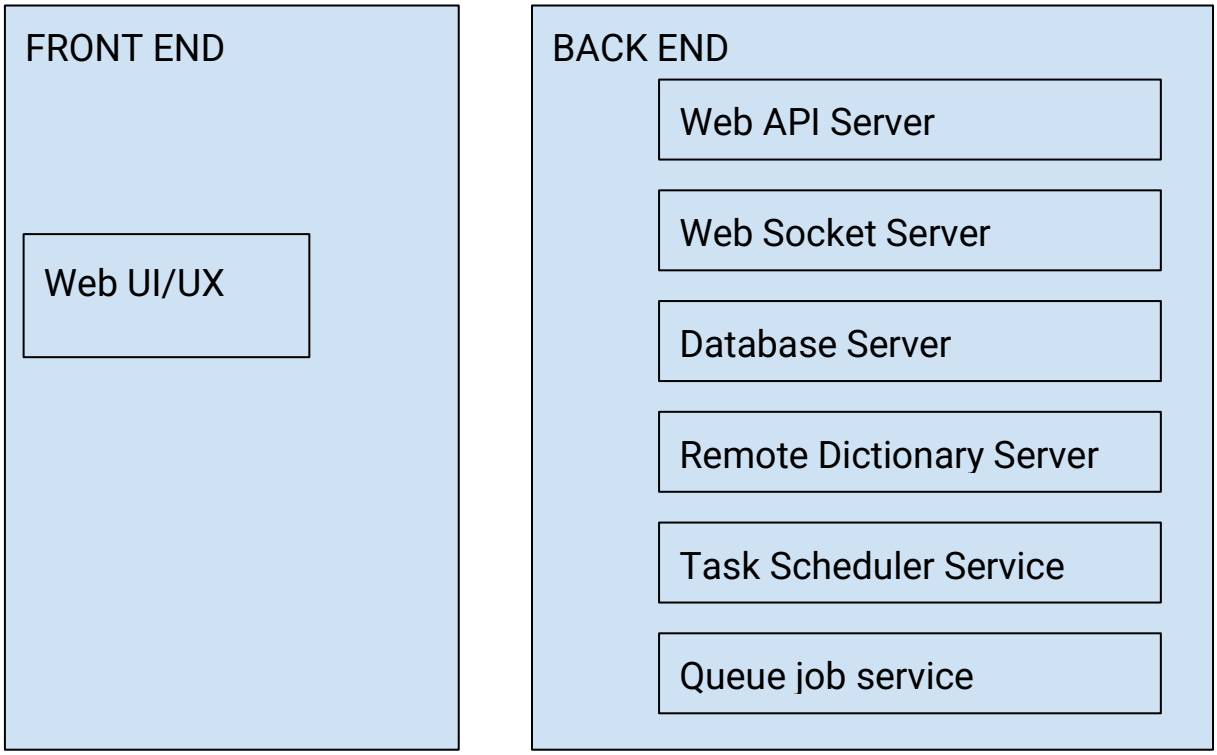
WebClient Components

Sentry WebClient architecture is divided into two core layers: a *Front End* and a *Back End*, each hosting dedicated components that work together to deliver a reliable and scalable user experience.

The *Front End* consists of the **Web UI/UX**, which serves as the primary interface through which users interact with the system via a web browser.

The *Back End* comprises six components. The **Web API Server** handles incoming client requests and orchestrates business logic. The **Web Socket Server** manages real-time, bidirectional communication between the client and server. The **Database Server** is responsible for persistent data storage and retrieval. The **Remote Dictionary Server** provides fast, in-memory data caching to optimise performance. The **Task Scheduler Service** manages the execution of time-based and recurring operations. Finally, the **Queue Job Service** handles asynchronous job processing, ensuring efficient distribution of background workloads.

Together, these components form a modular architecture designed for operational efficiency, maintainability, and a seamless user experience.



Prepare Installation

To begin the installation process, you will need to download *Sentry WebClient Installers* under latest Release available.

The Sentry software can be downloaded from the official Download Portal. Navigate to the portal, locate the product for which you need the installation files, and select the latest available release. After downloading the software, ensure it is saved and readily accessible on the appropriate servers for installation.

Installation steps

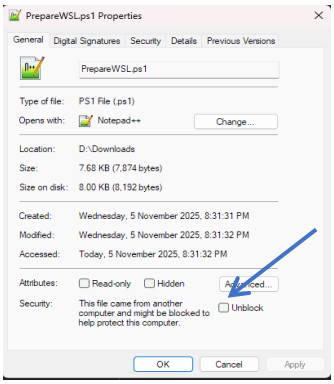
Windows Installation

> **INFORMATION**
>
> **Sentry WebClient** operates within a Docker container on top of Linux. If you are installing on a Windows OS, you will need to install Windows Subsystem for Linux (WSL) to enable the environment, as if the system were running on a native Linux machine.

Pre-Requisites

The following requisites needs to be validated before the installation process.

- Windows 11 24H2 or Windows Server 2022 or later.
- At least 100GB free in your disk to hold WSL + **WebClient** installation and database.
- Availability of *PrepareWSL.ps1* script file. Check that files are not Blocked. If it's, please unlock it.



> **IMPORTANT NOTE**
>
> If you have an old version of WSL or have Docker for Windows on the target machine, you need to uninstall first.

> **INFORMATION**
>
> WSL provides a Linux environment suitable for the Linux Docker images on a Windows machine. Once WSL is configured on your machine you can use docker to manage the services. **The ports opened by docker need to be made available on your Windows machine** – this process is described below.

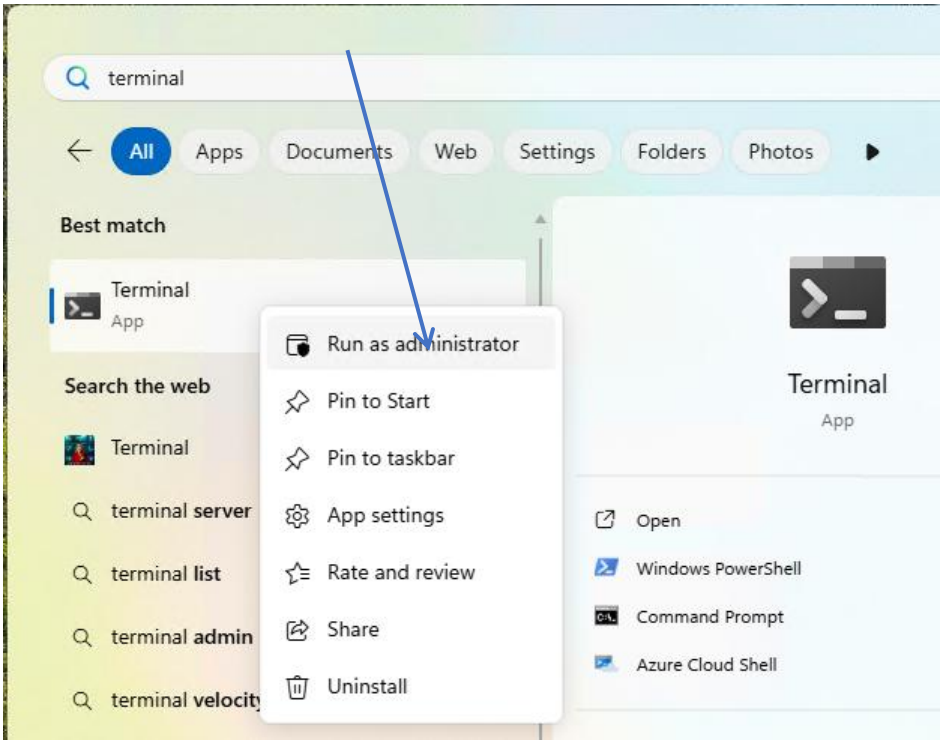
Install on a New Machine

The ***PrepareWSL.ps1*** script helps you configure WSL on a Windows machine.

It performs the following tasks:

- Enables WSL and configures the environment.
- Configures Windows to start WSL at boot time.
- Maps specified TCP ports from WSL to Windows and adds a Firewall rule to allow remote connections.
    - The tool requires you to specify which ports you want open. Check they are correct. If you need to open more ports in the future the tool has an alternative mode for that. For more detail, go to Adding WebClient if WSL is already installed and working.

To run the script, start *Powershell* as Administrator by searching for "Terminal" right-clicking and choosing "Run as administrator":

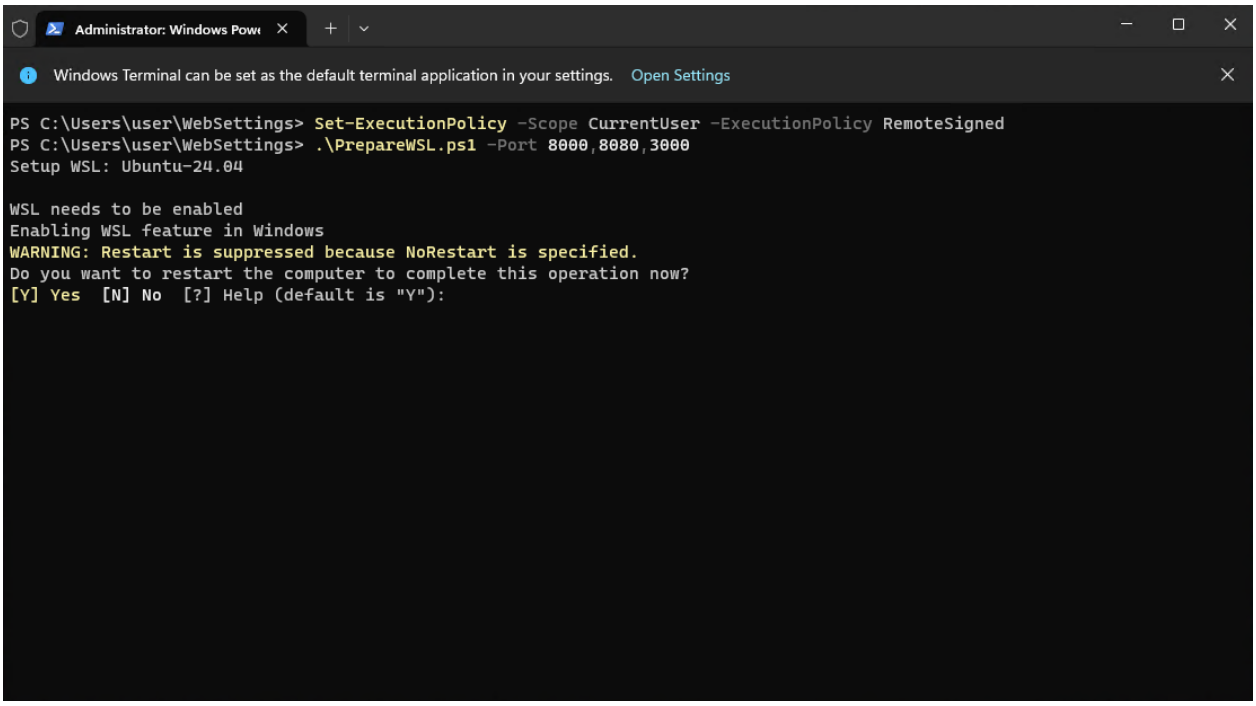


> **IMPORTANT NOTE**
>
> On a new machine you'll have to allow running PowerShell scripts.

> **INFORMATION**
>
> Make sure you know which ports you need to open. We're using the default ports **8000**, **8080** and **3000**.

```
> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
> .\PrepareWSL.ps1 -Port 8000,8080,3000
```

The first time you run the script – it's likely some components will require a reboot. Hit **Y** to do so when asked.



When the computer has rebooted restart PowerShell as above, return to where the script was run. Run it again to complete the installation.

You will be prompted part way through to create a username and password for your WSL account. You may remove the example username and write your own.

> **IMPORTANT NOTE**
>
> Please ensure that you remember your password, as it is essential for administrating your WSL instance without difficulty.

```
Starting...
Provisioning the new WSL instance Ubuntu-24.04
This might take a while...
Create a default Unix user account: sentry
New password:
Retype new password:
passwd: password updated successfully

Ubuntu-24.04 seemed to start correctly.
```

The script will continue after you confirm your password.

Here is the full output:

```
>.\PrepareWSL.ps1 -Port 8000,8080,3000
Setup WSL: Ubuntu-24.04
```

```
WSL is enabled
Checking WSL version
Downloading: Windows Subsystem for Linux 2.6.1
Installing: Windows Subsystem for Linux 2.6.1
Windows Subsystem for Linux 2.6.1 has been installed.
The operation completed successfully.
Checking for updates.
The most recent version of Windows Subsystem for Linux is already installed.
Installing Ubuntu-24.04
Downloading: Ubuntu 24.04 LTS
Installing: Ubuntu 24.04 LTS
Distribution successfully installed. It can be launched via 'wsl.exe -d Ubuntu-
24.04'
Configuring Ubuntu-24.04
Conversion in progress, this may take a few minutes.
The operation completed successfully.

Ubuntu-24.04 is ready, you may need to perform initial configuration

Starting...
Provisioning the new WSL instance Ubuntu-24.04
This might take a while...
Create a default Unix user account: sentry
New password:
Retype new password:
passwd: password updated successfully

Ubuntu-24.04 seemed to start correctly.

Configure Startup
Configured task "WSL Startup"

Ok.

Ok.

Ok.

Listen on ipv4:             Connect to ipv4:

Address         Port        Address         Port
--------------- ----------  --------------- ----------
0.0.0.0         8000        172.28.224.38   8000
0.0.0.0         8080        172.28.224.38   8080
0.0.0.0         3000        172.28.224.38   3000

Rule Name:                            WSL Port 8000
----------------------------------------------------------------------
Enabled:                              Yes
Direction:                            In
Profiles:                             Domain,Private,Public
Grouping:
LocalIP:                              Any
RemoteIP:                             Any
```

```
Protocol:                             TCP
LocalPort:                            8000
RemotePort:                           Any
Edge traversal:                       No
Action:                               Allow
Ok.

Rule Name:                            WSL Port 8080
----------------------------------------------------------------------
Enabled:                              Yes
Direction:                            In
Profiles:                             Domain,Private,Public
Grouping:
LocalIP:                              Any
RemoteIP:                             Any
Protocol:                             TCP
LocalPort:                            8080
RemotePort:                           Any
Edge traversal:                       No
Action:                               Allow
Ok.

Rule Name:                            WSL Port 3000
----------------------------------------------------------------------
Enabled:                              Yes
Direction:                            In
Profiles:                             Domain,Private,Public
Grouping:
LocalIP:                              Any
RemoteIP:                             Any
Protocol:                             TCP
LocalPort:                            3000
RemotePort:                           Any
Edge traversal:                       No
Action:                               Allow
Ok.
```

You can find more details about what is being done in the sections Setting WSL to start at boot without requiring a login and Opening ports for using WebClient.

Adding WebClient if WSL is already installed and working

If you already have WSL installed for some other purpose, you may use the **PrepareWSL.ps1** script to perform some specific helpful tasks.

- Setting WSL to start at boot without requiring a login.
- Opening ports for using WebClient.

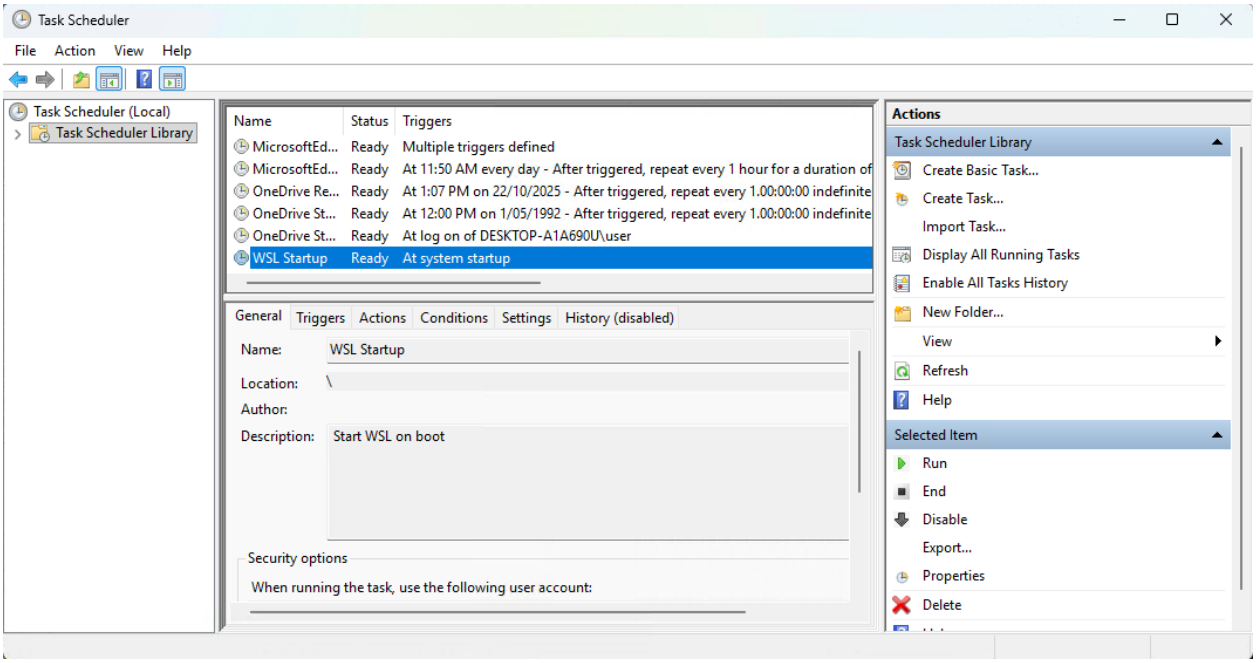
Setting WSL to start at boot without requiring a login

By default, WSL does not automatically start unless you log in and run something. ***PrepareWSL.ps1*** can configure this for you.

> **INFORMATION**
>
> In case another software product might manage WSL to start for you at boot time already, skip this section.

```
>\PrepareWSL.ps1 -ConfigureStartup
Configure Startup
Configured task "WSL Startup"
```

This configures a job with Task Scheduler to start WSL on boot. You can change or remove this task in Task Scheduler.



Opening ports for using WebClient

By default, TCP ports open in WSL are not available on the host Windows machine's IP address.

PrepareWSL.ps1 can configure this for you. You'll need to know which ports you will be using with WebClient. We're using the default ports **8000**, **8080** and **3000**.

```
>.\PrepareWSL.ps1 -Redirect -Port 8000,8080,3000
```

In the normal WSL networking mode of operation "NAT" we need to configure port-forwarding rules to bind the ports externally. On other network modes of operation this is not needed so you might not see this output:

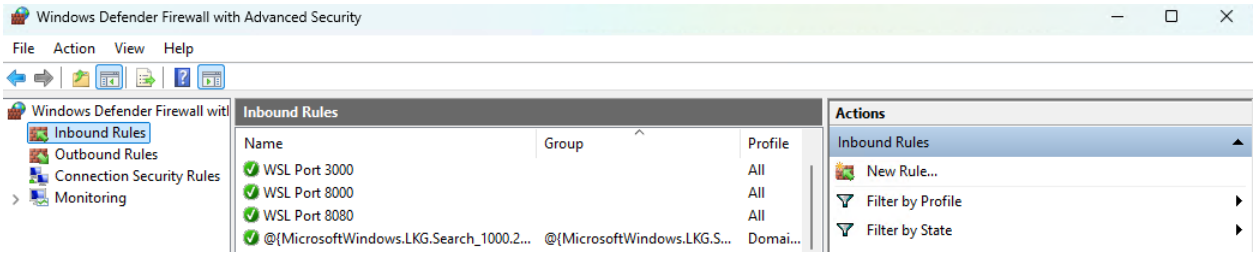
```
Listen on ipv4:             Connect to ipv4:

Address         Port        Address         Port
--------------- ----------  --------------- ----------
0.0.0.0         8000        172.28.224.38   8000
0.0.0.0         8080        172.28.224.38   8080
0.0.0.0         3000        172.28.224.38   3000
```

This shows that the host Windows machine will connect any external IP (denoted by 0.0.0.0) to the internal WSL IP port of the same number.

Additionally, to allow remote hosts to contact **Sentry WebClient** we add firewall rules to allow incoming connection on these ports. The following report is repeated for all ports specified with -Port:

```
Rule Name:                            WSL Port 8000
----------------------------------------------------------------------
Enabled:                              Yes
Direction:                            In
Profiles:                             Domain,Private,Public
Grouping:
LocalIP:                              Any
RemoteIP:                             Any
Protocol:                             TCP
LocalPort:                            8000
RemotePort:                           Any
Edge traversal:                       No
Action:                               Allow
Ok.
```

This shows that Inbound TCP connections on LocalPort 8000 will be accepted. The `Rule Name` can be used to find and change the rule in the Windows Firewall.



Installing Web Client on WSL

After the script has completed you will have a working Ubuntu Linux container on your Windows machine.

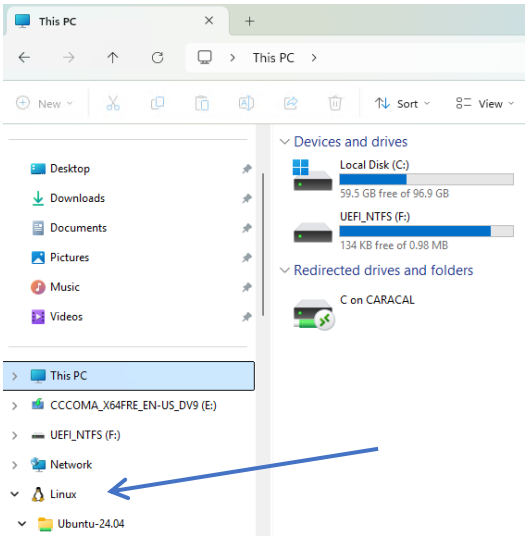
You can enter your WSL instance by simply typing **wsl**.

```
PS C:\Users\user\WebSettings> wsl
sentry@DESKTOP-A1A690U:/mnt/c/Users/user/WebSettings$
```

By default, you will be in the same directory you ran the `wsl` command from. You can access files from your Windows disk under **/mnt/c** and to find a Windows directory in your Linux container you can use the `wslpath` tool.

```
$ wslpath "c:\ProgramData"
/mnt/c/ProgramData
```

Alternatively, you will find your WSL Linux filesystem available in explorer:



You should be able to open and change most files normally by using explorer – or accessing the path **`\\wsl.localhost\Ubuntu-24.04\home\sentry`**. – e.g. for **`/home/sentry`**.

> **INFORMATION**
>
> To configure and deploy the **Sentry WebClient** please follow the remaining steps for a native Linux installation in section Sentry WebClient Deployment.

> **IMPORTANT NOTE**
>
> The first time you configure WSL with ***PrepareWSL.ps1*** you will be able to configure and run using the wsl command. However, since you have not rebooted WSL will not be running in the background if you exit the wsl shell or close the window. This is normal for WSL but probably not what you want. After you have rebooted this issue will not occur again.

Sentry WebClient Deployment

> **INFORMATION**
>
> The instructions in this section apply to both native Linux operating systems and Windows systems where WSL (Windows Subsystem for Linux) has been installed and configured, as described in earlier sections. This ensures users on either platform can follow the deployment steps seamlessly.

Install Docker Engine in Ubuntu

From a terminal inside WSL or a native Linux OS, first install Docker.

```
$ sudo apt update
$ sudo apt install docker.io docker-compose-v2
$ sudo adduser $USER docker
```

Log out and log back in, or run:

```
$ newgrp docker
```

Sanity check: Test Docker:

```
$ docker run hello-world
```

Initial Steps

1. Login into our Docker repository. Run and paste your provided token:

```
$ docker login docker.io -u isentryintelex

i Info → A Personal Access Token (PAT) can be used instead.
         To create a PAT, visit https://app.docker.com/settings

Password:

WARNING! Your credentials are stored unencrypted in
'/home/user/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
```

2. Copy into WSL from Windows the zip holding docker files.

> **IMPORTANT NOTE**
>
> **Do not unzip in Windows and copy to WSL as you might encounter encoding issues.**

```
user@DESKTOP-1303Q2Q:~$ scp /mnt/c/Users/user/Downloads/isentrywebclient-docker.zip .
```

You might need to install unzip in your WSL (`sudo apt install unzip`). It is recommended to copy into WSL with (path is example):

3. Make your own **.env** file, using **.env.example** as a reference. You can cp .env.example into .env like:

```
user@DESKTOP-1303Q2Q:~/isentrywebclient-main$ cp .env.example .env
```

Advanced Steps

1. Download these images on Dockerhub or make sure they are set in .env file.

> **INFORMATION**
>
> This step is optional given the docker engine will auto pull it if they are missing.

```
REGISTRY_NGINX=nginx:latest
REGISTRY_LARAVEL= comintelexvision/sentry-client-api:2026R1
REGISTRY_DB=mariadb:10.11
REGISTRY_NODE=comintelexvision/sentry-client-node:2026R1
REGISTRY_CMS= comintelexvision/sentry-client-cms:2026R1
```

2. Copy file **`.env.example`** to file **.env** and change **.env** file.

> **INFORMATION**
>
> Below is an example of an entire **.env** file.

```
REGISTRY_NGINX=nginx:latest
REGISTRY_LARAVEL= comintelexvision/sentry-client-api:2026R1
REGISTRY_DB=mariadb:10.11
REGISTRY_NODE=comintelexvision/sentry-client-node:2026R1
REGISTRY_CMS= comintelexvision/sentry-client-cms:2026R1
```

```
REGISTRY_PHPMYADMIN=phpmyadmin/phpmyadmin:5.2.0
REGISTRY_REDIS=redis:7.0.4

FOLDER_VOLUME_DATA=/home/user/webclient/data

DB_HOST=db-camera
DB_ROOT_PASSWORD=12345678
DB_DATABASE=db_intelexvision
DB_USERNAME=user_intelexvision
DB_PASSWORD=12345678
DB_PORT=3306

PHPMYADMIN_PORT=1000

REDIS_PASSWORD=12345678
REDIS_DB_PUSB=0
REDIS_PREFIX=laravel_database_psub:
REDIS_PORT=6379
NODE_PORT=3000

PUSH_DATA_USERNAME=ghnckmfFQgfdnjnv?pg
PUSH_DATA_PASSWORD=fd9<gbaqkrhxjhgmb04/

CMS_PORT=8080
NGINX_PORT=8000

# added 19 May 2023 to limit resources
MAX_API_CPUS=2.0 # two cores
MAX_API_MEMORY=8000M
MAX_DB_CPUS=2.0
MAX_DB_MEMORY=8000M

# 11/02/2025 node js config
# https mode HTTPS_MODE 0: OFF, 1: ON
SSL_KEY_PATH=""
SSL_CERT_PATH=""
HTTPS_MODE=0

# telegram token
TELEGRAM_API_TOKEN="8147021175:AAHRG0lK4mwbHfrEHmQMyeXOGRobeWCV6UY"
NODE_COMMAND='node index.js & node bot-telegram.js'

### example
# rabbit mq server info for keep watch
# RABBIT_MQ_HOST=192.168.30.228
# RABBIT_MQ_PORT=5672
# RABBIT_MQ_USER=user
# RABBIT_MQ_PASSWORD=password
# RABBIT_MQ_VHOST=intelex
# RABBIT_MQ_EXCHANGE=intelex_exchange
# RABBIT_MQ_QUEUE_TO_SEND=iv_keepwatch_test2_queue
# RABBIT_MQ_ROUTING_TO_SEND=iv_keepwatch_test2
# RABBIT_MQ_QUEUE_TO_RECEIVE=iv_keepwatch_test_queue
```

```
# RABBIT_MQ_ROUTING_TO_RECEIVE=iv_keepwatch_test

RABBIT_MQ_HOST=
RABBIT_MQ_PORT=5672
RABBIT_MQ_USER=
RABBIT_MQ_PASSWORD=
RABBIT_MQ_VHOST=
RABBIT_MQ_EXCHANGE=
RABBIT_MQ_QUEUE_TO_SEND=
RABBIT_MQ_ROUTING_TO_SEND=
RABBIT_MQ_QUEUE_TO_RECEIVE=
RABBIT_MQ_ROUTING_TO_RECEIVE=

QUANTITY_RECORD_AUTO_DELETE=50
DISK_SIZE_NEED_BLANK=100
MAIL_MAILER=smtp
MAIL_HOST=mailhog
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="ISentry Web Client"

## example
# MAIL_MAILER=smtp
# MAIL_HOST=smtp.gmail.com
# MAIL_PORT=587
# MAIL_USERNAME=example_user@intelexvision.com
# MAIL_PASSWORD="1234 5678 bdnk hzcy"
# MAIL_ENCRYPTION=tls
# MAIL_FROM_ADDRESS=example_user@intelexvision.com
# MAIL_FROM_NAME="ISentry Web Client"
##
FRONT_END_ALLOWED_ORIGIN_URL='http://192.168.1.3:8080,https://192.168.1.3:8080'
##
```

> **IMPORTANT NOTE**
>
> Pay attention to the following information to limit your resources accordingly.

```
# limit resources
MAX_API_CPUS=2.0 # two cores
MAX_API_MEMORY=8000M
MAX_DB_CPUS=2.0
MAX_DB_MEMORY=8000M
```

3. Run script to prepare volume data directory.

Sentry Web Client requires some preset folder structure in FOLDER_VOLUME_DATA set in **.env** file. Do it by run with a bash terminal like:

```
./docker-chmod-folder.sh
```

Verify folder structure of FOLDER_VOLUME_DATA:

```
├──────api
│     ├──────bootstrap
│     │     └──────cache
│     ├──────download
│     ├──────excel
│     ├──────group-cameras
│     ├──────images
│     │     ├──────medium
│     │     └──────small
│     ├──────pdf
│     └──────storage
│           ├──────app
│           ├──────backup
│           ├──────debugbar
│           ├──────framework
│           │     ├──────cache
│           │     ├──────laravel-excel
│           │     ├──────sessions
│           │     ├──────testing
│           │     └──────views
│           └──────logs
```

4. From terminal run…

```
./start.sh
```

The normal output is shown below.

```
sentry@MSI:~/isentrywebclient-2026R1-26.01.21$ ./start.sh
HTTPS_MODE is: 0
HTTPS mode is disabled
WARN[0000] /home/sentry/isentrywebclient-2026R1-26.01.21/docker-compose.yml: the attribute
`version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 6/6
 ✓ Container db-camera       Healthy                    0.8s
 ✓ Container cms-camera      Running                    0.0s
 ✓ Container redis-camera    Running                    0.0s
 ✓ Container node-camera     Running                    0.0s
 ✓ Container laravel-camera  Started                    1.0s
 ✓ Container nginx-camera    Running
```

You can check that all the containers are up and running by entering the command:

```
sentry@MSI:~/isentrywebclient-2026R1-26.01.21$ docker ps
CONTAINER ID   IMAGE                                                                    COMMAND
CREATED        STATUS                  PORTS                                          NAMES
d263745ed695   docker-registry.au.intelexvision.com/sentry-client-api:v20260130-035848.81   "docker-php-
entrypoi…"   16 hours ago   Up 16 hours             9000/tcp
laravel-camera
9d380c7a605d   docker-registry.au.intelexvision.com/sentry-client-cms:v20260128-002045.51   "/docker-
entrypoint.…"   16 hours ago   Up 16 hours             0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
cms-camera
796df01b6b4a   nginx:latest                                                             "/docker-
entrypoint.…"   17 hours ago   Up 17 hours             0.0.0.0:8000->80/tcp, [::]:8000->80/tcp
nginx-camera
```

```
a14a655ea442   comintelexvision/sentry-client-node:2026R1                                  "docker-
entrypoint.s…"   17 hours ago   Up 17 hours            0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp
node-camera
1dee31558234   redis:7.0.4                                                                 "docker-
entrypoint.s…"   17 hours ago   Up 17 hours            6379/tcp
redis-camera
cb0f5f8231ab   mariadb:10.11
```

> **IMPORTANT NOTE**
>
> *For CORS Issue*
>
> When accessing from computer IP you need *FRONT_END_ALLOWED_ORIGIN_URL* properly configured. Update the variable in your **.env** file accordingly, for example below we accept connections from IP 192.168.1.3 in HTTP and HTTPS.
>
> ```
> FRONT_END_ALLOWED_ORIGIN_URL='http://192.168.1.3:8080,https://192.168.1.3:8080'
> ```

Upgrade Process

Simple edit the image registry tag for the updated version.

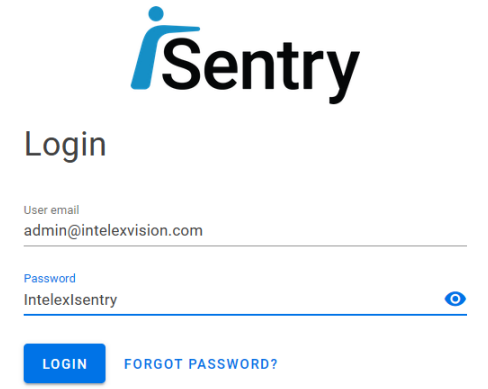
Post Installation

Accessing the Web Client

Use a web browser and type http://x.x.x.x:8080 with x.x.x.x is **Sentry WebClient** Server IP address. Make sure the machine running the web browser can see the **Sentry WebClient** Server. If web browser and **Sentry WebClient** Server are on the same machine, the address http://localhost:8080 can be used to access.

> **INFORMATION**
>
> Default Login Credentials are as follows:
>
> • **Username:** admin@intelexvision.com
> • **Password:** IntelexSentry

> **IMPORTANT NOTE**
>
> For your security, it is strongly recommended to change the default username and password immediately after your first login. Using unique credentials helps protect your system from unauthorized access and potential security threats. Make sure to choose a strong password that combines letters, numbers, and special characters.


iSentry
Login
User email
admin@intelexvision.com
Password
IntelexIsentry
LOGIN
FORGOT PASSWORD?

Troubleshooting and Common Issues

Common Issues

Database Restoration

Rarely, events like power outages, between others, can corrupt the MYSQL database beyond recovery.

This section explains how to restore your Database using one of the automatic daily backups. These backup files are stored in the following folder:

`FOLDER_VOLUME_DATA/api/storage/backup`

The location `FOLDER_VOLUME_DATA` is a shortcut that points to the main data directory, which is defined in your system's environment settings file **.env**. For example:

```
FOLDER_VOLUME_DATA=/home/user/webclient_data
```

With this setting, your actual backup files can be found at:
`/home/user/webclient_data/api/storage/backup`.

Knowing this path will help you locate and restore your backups whenever needed.

Restore database using script method (highly recommended)

From the location of the deployment script folder, run **`ls`** to make sure the script **`restore webclient_db.sh`** is available in the folder.

```
user@inspriron-15:~/Work/isentrywebclient$ ls
db-camera               docker-compose.https.yml  images  README.md                start-
intelexvision.sh  stop.sh
docker-chmod-folder.sh  docker-compose.yml        nginx   restore_webclient_db.sh  start.sh
```

Make sure the script can be executed using `chmod +x` command.

```
user@inspriron-15:~/Work/isentrywebclient$ chmod +x restore_webclient_db.sh
```

Run the restore script.

```
user@inspriron-15:~/Work/isentrywebclient$ ./restore_webclient_db.sh
FOLDER_VOLUME_DATA: /home/user/webclientdata
Backup directory  : /home/user/webclientdata/api/storage/backup
Data directory    : /home/user/webclientdata/data-db

This will REPLACE the MySQL data directory. Continue? (y/N)
```

Type 'y' to continue, another confirmation question will appear to

```
Newest base file :
/home/user/webclientdata/api/storage/backup/base_tables_2025_12_15_02_00_01.sql
Alert files      :
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_11_30_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_01_02_00_01.sql
```

```
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_02_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_03_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_03_11_58_12.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_03_12_00_58.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_04_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_05_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_06_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_07_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_08_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_09_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_10_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_11_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_12_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_13_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_14_02_00_01.sql
  /home/user/webclientdata/api/storage/backup/alert_tables_2025_12_15_02_00_01.sql

Proceed with these backup files? (y/N)
```

Continue to press ‘y’ to keep going.

```
Stopping all web client containers...
[+] Running 7/7
 ✓ Container cms-camera      Removed
 ✓ Container node-camera     Removed
 ✓ Container nginx-camera    Removed
 ✓ Container laravel-camera  Removed
 ✓ Container redis-camera    Removed
 ✓ Container db-camera       Removed
 ✓ Network network-camera    Removed
Renaming existing data directory...
[sudo] password for user:
```

The script is going to rename the old broken database folder and require root permission to keep going. The restoring process will take a while depending on how large the database size is.


```
Stopping all web client containers...
[+] Running 7/7
 ✓ Container cms-camera      Removed
 ✓ Container node-camera     Removed
 ✓ Container nginx-camera    Removed
 ✓ Container laravel-camera  Removed
 ✓ Container redis-camera    Removed
 ✓ Container db-camera       Removed
 ✓ Network network-camera    Removed
Renaming existing data directory...
[sudo] password for user:
```

The script is going to rename the old broken database folder and require root permission to keep going. The restoring process will take a while depending on how large the database size is.


```
Starting db-camera to create fresh database...
[+] Running 2/2
```

```
 ✓ Network network-cameraCreated 0.0s
 ✓ Container db-camera Started 0.2s
Starting laravel-camera to generate schema...
[+] Running 3/3
 ✓ Container db-camera Healthy 5.1s
 ✓ Container redis-cameraStarted 5.2s
 ✓ Container laravel-cameraStarted 5.3s
Preparing table template in db......
Stopping all services except db-camera...
[+] Stopping 1/1
 ✓ Container laravel-cameraStopped10.1s
Running restore inside temporary mariadb:10.11 container...
Restoring base tables from:
/home/user/webclientdata/api/storage/backup/base_tables_2025_12_15_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_11_30_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_01_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_02_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_03_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_03_11_58_12.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_03_12_00_58.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_04_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_05_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_06_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_07_02_00_01.sql
Restoring alerts from:
/home/user/webclientdata/api/storage/backup/alert_tables_2025_12_08_02_00_01.sql
```

Restore database manually (only when the script method does not work)

View list backup file with command ll or ls -al

```
[0] user@pangolin:~/webclient_data/api/storage/backup$ ll
total 442980
drwxr-xr-x 2 root root     4096 Dec 13 01:16 ./
drwxrwxr-x 7 user user     4096 Dec 13 01:16 ../
-rwxrwxrwx 1 user user 33917213 Dec  1 13:03 alert_tables_2025_12_01_02_00_03.sql*
-rwxrwxrwx 1 user user 26784510 Dec  2 13:03 alert_tables_2025_12_02_02_00_30.sql*
-rwxrwxrwx 1 user user 43273010 Dec  3 13:03 alert_tables_2025_12_03_02_00_05.sql*
-rwxrwxrwx 1 user user 66026110 Dec  6 13:03 alert_tables_2025_12_06_02_00_08.sql*
-rwxrwxrwx 1 user user 42284323 Dec  7 13:04 alert_tables_2025_12_07_02_00_08.sql*
-rwxrwxrwx 1 user user 43419402 Dec  8 13:02 alert_tables_2025_12_08_02_00_04.sql*
-rwxrwxrwx 1 user user 38791063 Dec  9 13:02 alert_tables_2025_12_09_02_00_04.sql*
-rwxrwxrwx 1 user user 44291454 Dec 10 13:02 alert_tables_2025_12_10_02_00_04.sql*
-rwxrwxrwx 1 user user 42488307 Dec 11 13:02 alert_tables_2025_12_11_02_00_06.sql*
-rwxrwxrwx 1 user user 47976162 Dec 12 13:00 alert_tables_2025_12_12_02_00_06.sql*
-rwxrwxrwx 1 user user  2414332 Dec  1 13:03 base_tables_2025_12_01_02_00_03.sql*
-rwxrwxrwx 1 user user  2415816 Dec  2 13:03 base_tables_2025_12_02_02_00_30.sql*
```

```
-rwxrwxrwx 1 user user  2415783 Dec  3 13:03 base_tables_2025_12_03_02_00_05.sql*
-rwxrwxrwx 1 user user  2433737 Dec  6 13:03 base_tables_2025_12_06_02_00_08.sql*
-rwxrwxrwx 1 user user  2433716 Dec  7 13:04 base_tables_2025_12_07_02_00_08.sql*
-rwxrwxrwx 1 user user  2433785 Dec  8 13:02 base_tables_2025_12_08_02_00_04.sql*
-rwxrwxrwx 1 user user  2424932 Dec  9 13:02 base_tables_2025_12_09_02_00_04.sql*
-rwxrwxrwx 1 user user  2441145 Dec 10 13:02 base_tables_2025_12_10_02_00_04.sql*
-rwxrwxrwx 1 user user  2441167 Dec 11 13:02 base_tables_2025_12_11_02_00_06.sql*
-rwxrwxrwx 1 user user  2456622 Dec 12 13:01 base_tables_2025_12_12_02_00_06.sql*
```

Rule to restore

- Restore only the newest base_tables_*.sql.
- Restore all alert_tables_*.sql after that.

Steps to restore

Stop all web client container. From script folder run.

```
[[0] user@pangolin:~/isentrywebclient$ ./stop.sh
[+] Stopping 6/6
✓ Container node-camera     Stopped            11.5s
✓ Container cms-camera      Stopped            11.5s
✓ Container nginx-camera    Stopped             1.1s
✓ Container laravel-camera  Stopped            11.4s
✓ Container redis-camera    Stopped             1.1s
✓ Container db-camera       Stopped
```

Rename `FOLDER_VOLUME_DATA/data-db` to e.g: `FOLDER_VOLUME_DATA/data-db-old`

```
[0] user@pangolin:~/webclient_data$ sudo mv data-db data-db-old
```

Start web client `camera-db`. This step help to create new **Sentry WebClient** database.

```
user@pangolin:~/isentrywebclient$ docker compose up -d db-camera
[+] Running 1/1
 ✓ Container db-camera  Started
```

Wait for about 5 seconds to make sure database server has been ready. Then start **Sentry WebClient** `laravel-camera` container. This step is to generate template for each table in database.

```
[0] user@pangolin:~/isentrywebclient$ docker compose up -d laravel-camera
[+] Running 3/3
 ✓ Container db-camera       Healthy                0.5s
 ✓ Container redis-camera    Started                0.8s
 ✓ Container laravel-camera  Started                1.1s
```

Wait for about 60 seconds to make sure all table template has been created, then stop `laravel-camera`. Except `camera-db` container, other web client pieces must be stopped. This step to avoid incoming alert will break the database during restoration.

```
user@pangolin:~/isentrywebclient$ docker compose stop laravel-camera
[+] Stopping 1/1
 ✓ Container laravel-camera  Stopped
```

After steps above, the Database structure has been prepared, now is the time to restore backup file using mysql command line.

Example of restoring file name `base_tables_2025_12_12_02_00_06.sql`

```
db_file=base_tables_2025_12_12_02_00_06.sql && \
docker run --rm -it \
  -v /home/user/webclient_data/api/storage/backup:/backup_data \
  --network network-camera \
  mariadb:10.11 bash -c "
    set -e
    echo 'Restoring base tables from: $db_file'
    mysql -h db-camera -u root -p12345678 db_intelexvision <
/backup_data/$db_file
  "
```

Repeat command above with different `sql` files. Make sure to check the Rule to restore section to restore the backed sql file in a correct order.

Start all **Sentry WebClient** containers and try to login to **Sentry WebClient** to verify that database has been restore.

Stop Sentry WebClient

Simply run ./stop.sh command and it will trigger the process to stop **Sentry WebClient** server.

WSL

1. If you do not see a camera in the **Sentry WebClient** *Alert View* or *Past Alerts* you might double check under permission for your user to see whether it is allowed there.

2. If you forget the Ubuntu password you can run the following from a PowerShell with Administrator rights and run again `PrepareScript.ps1` script.

```
wsl --unregister Ubuntu-24.04
```

3. If you run out of space on the disk you can migrate the virtual disk. You can run from PowerShell (not admin or might have trouble later):

```
wsl --manage Ubuntu-24.04 --move D:\some\directory
```

The original location of the virtual disk is in **`%LOCALAPPDATA%\Packages`** and in that location search for `CanonicalGroupLimited.Ubuntu_2204` and folder `LocalState`.

CORS issue

When accessing from computer IP you need `FRONT_END_ALLOWED_ORIGIN_URL` properly configured. Update the variable in your **.env** file accordingly, for example below we accept connections from IP 192.168.1.3 in HTTP and HTTPS.

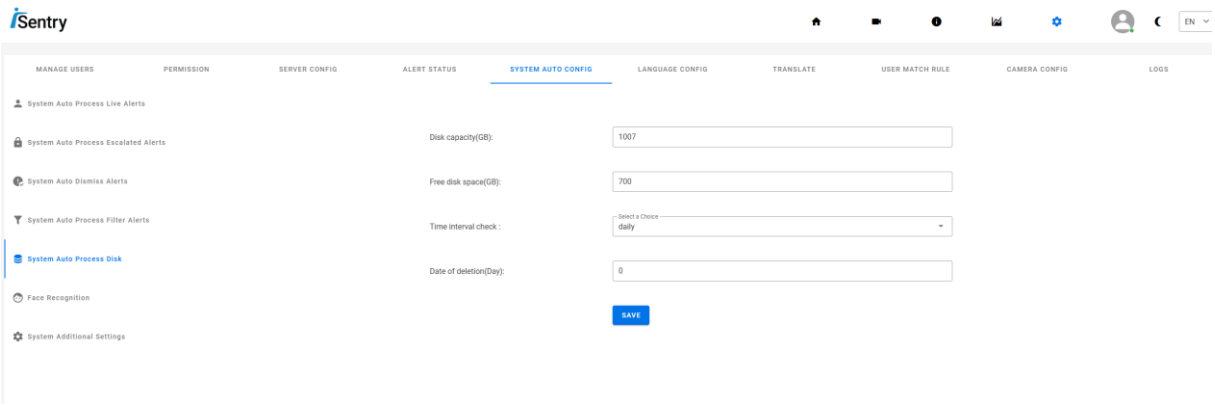
```
FRONT_END_ALLOWED_ORIGIN_URL='http://192.168.1.3:8080,https://192.168.1.3:8080'
```

Deletion issue on WSL

The default size of the image disk in WSL is 1TB, you can double check by running the command:

```
sentry@MSI:/mnt/c/Users/Victor$ df -Th /
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/sdd       ext4 1007G   42G  914G   5% /
```

The command says in this example that the disk size is **1007 GB** and **914 GB** are available, so if we want 200GB for our **Sentry Web Client** we can enter as follows (`914 GB – 200 GB = 714GB ~ 700GB`).



Support Information

If you need Technical Support with Intelex Vision systems, please write an email to customerservices@intelexvision.com and we will certainly help to solve the problem.