Kiuwan On-Premises Distributed Installation Guide

- Introduction
- System requirements
  - Installation requirements
    - Special requirements for RedHat, CentOS, and Fedora
  - Needed internet connections
  - Proxy configuration
    - Examples
  - CPU and memory minimum requirements
- The Kiuwan On-Premises installation tool (kiuwan-cluster)
- Installation: common steps
  - Step 1: download kiuwan-cluster
  - Step 2: untar kiuwan-cluster
  - Step 3: copy license files
  - Step 4: download and copy the needed driver version for MySQL
  - Step 5: configure the volumes paths
  - Step 6: initialize your volumes
  - Step 7: configure your email server
    - Mail server configuration examples
- Installation: single-host and minimum configuration
  - Step 1: deploy user content
  - Step 2: install Kiuwan On-Premises
- Accessing your Kiuwan On-Premises installation
  - Step 1: add your domain to your local network DNS
  - Step 2: add your certificates' CA to your clients
  - Step 3: access Kiuwan On-Premises
    - Accessing the web application
    - Downloading Kiuwan Local Analyzer
    - Consuming Kiuwan REST API
    - Configuring Kiuwan for Developers
  - Default users

Introduction

The Kiuwan On-Premises installer is a powerful tool that suits multiple environment scenarios:

- Single-host fully contained installation.
- Multi-host installation where infrastructure services are externalized.
- Multi-host installation where Kiuwan services are distributed.
- Any combination of the two previous scenarios.

Depending on your needs, a different installation approach will be needed. Check this installation guide for details on how to proceed and to find the solution that best fits your requirements.

System requirements

Installation requirements

It is mandatory for any host where Kiuwan On-Premises is installed to meet these requirements:

- Linux distribution
- Linux kernel version 3.10 or higher
- Connectivity to SMTP Mail Server.
- Internet connectivity during the installation process (see Needed internet connections).
- Installing user must be root or have sudo privileges.
- When running the provided Redis Cluster containers (that is, if you are not externalizing the Redis service, see Externalizing Redis), it is recommended for the host machine to keep the same local IP address. If this is not acceptable for you, be aware that all containers should be restarted after an IP change (see My Kiuwan On-Premises local IP address has changed and Redis is not responding anymore).



Please follow Docker official recommendations when installing Docker. These URLs describe the installation process for different Linux distributions:

- https://docs.docker.com/install/linux/docker-ce/centos/
- https://docs.docker.com/install/linux/docker-ce/debian/
- https://docs.docker.com/install/linux/docker-ce/fedora/
- https://docs.docker.com/install/linux/docker-ce/ubuntu/

These software are also needed:

- Docker CE >= **19.03.2**
- Docker-compose >= **1.24.1**
- Unzip
- GNU tar
- Wget
- BC
- Tee
- Ip
- Java Runtime Environment >= **8** (needed to generate keystores for custom hosts).
- OpenSSL >= **1.1.1** (needed to generate certificates for custom hosts). Other supported versions of OpenSSL are **1.0.2g** and **1.0.2k-fips**.

Make sure that the docker-compose command can be executed with sudo. This means that the docker-compose binary file must be included in the sudoers path. The binary is usually installed under the /usr/local/bin directory. Depending on your OS, this information may be set in the secure_path variable in the sudoers file (/etc/sudoers). Please check the official Docker-compose documentation for more information: https://docs.docker.com/compose/install.

It is also recommended to use the target installation hosts **exclusively** for Kiuwan services. If you plan on running other containers than Kiuwan's in a single-host installation, please make sure that:

- None of them is using the following network: 172.172.0.0/16
- There are no containers that may block any of these ports:
  - 443: standard HTTPS port (or the one you will configure to access your Kiuwan On-Premises installation).
  - 8143, 8243, 8343, 8443, 8543, 8643, 8743, 8843: ports that Kiuwan front instances listen to (or those you may configure for Kiuwan front instances).
  - 6379: port that Redis instances listen to.
  - 3306: port that MySQL instance listen to (or the port you configure for MySQL).

The Kiuwan On-Premises installation tool will check most of these requisites before installing. If any of them is not met, installation will be canceled.

Special requirements for RedHat, CentOS, and Fedora

If your docker server is running on RedHat, CentOS or Fedora, be sure the filesystem where docker is installed supports d_type (the directory entry data structure that describes the information of a directory on the filesystem).

Some of the above operating systems are not configured with d_type support (see http://www.pimwiddershoven.nl/entry/docker-on-centos-7-machine-with-xfs-filesystem-can-cause-trouble-when-d-type-is-not-supported.

Running on XFS without d_type support causes Docker to skip the attempt to use the overlay or overlay2 driver. See https://docs.docker.com/storage/storagedriver/overlayfs-driver/#prerequisites.

You can check if your existing XFS filesystem has d_type enabled by running the following commands:

```
$ docker info | grep "Supports d_type:"
Supports d_type: true

$ xfs_info /docker-mount-point | grep ftype
naming =version 2         bsize=4096      ascii-ci=0     ftype=1
```

In case you get d_type: false or ftype=0, you will need to create a new XFS filesystem with d_type support enabled. Unfortunately, it isn't possible to enable d_type support on an existing filesystem.

The Kiuwan On-Premises installation tool will check if d_type is enabled on XFS filesystems and stop the installation if it is not.

There are two options:

- Add a new disk and create a new XFS partition on it.
- Backup your existing data and recreate the XFS filesystem with d_type support enabled.

You can create a new XFS filesystem with d_type enabled by running the following command:

```
$ mkfs.xfs -n ftype=1 /mount-point
```

Needed internet connections

Please make sure your host machines have connection to these servers when installing Kiuwan On-Premises:

Host	Needed when	Purpose
*.docker.com (all subdomains of docker.com, including nested subdomains) *.docker.io (all subdomains of docker.io)	Installing	These are the Docker servers where the needed images will be pulled from.
cdn.mysql.com	Installing	Download the mysql driver file needed during the installation process.
static.kiuwan.com	Installing	This is Kiuwan's static content server, needed by the installer to download needed resources.
api.kiuwan.com	You own a Kiuwan On-Premises **Insights** license, both for installing and running	This is Kiuwan's central API endpoint, needed to update Insights vulnerabilities database.

Proxy configuration

If the host on which you are installing Kiuwan On-Premises needs to access the internet through a proxy server, note that:

- Your OS should know about your proxy settings in order to be able to download the Kiuwan On-Premises installer distribution.
- Docker daemon must know about your proxy settings in order to download images from https://*.docker.com or https://*.docker.io
- Docker must know about your proxy settings in order to propagate them to the created containers.
- Direct access to your SMPT mail server is needed (it is **MANDATORY** that the host machine can access your mail server regardless of the proxy server).

To instruct your OS to use your proxy settings, please refer to the official documentation for your Linux distribution.

To make Docker daemon use your proxy server, please follow the official Docker documentation on how to set a proxy:

https://docs.docker.com/config/daemon/systemd/#httphttps-proxy

To make Docker use your proxy when creating containers, please follow the official Docker documentation on how to set a proxy to be used by the containers:

https://docs.docker.com/network/proxy/#configure-the-docker-client

Remember to restart both docker daemon and docker to apply proxy configuration changes:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

You can check if Docker has successfully read your configuration by executing:

```
docker system info
```

Check if your proxy configuration is shown in the console:

```
[...]
HTTP Proxy: http://proxy.domain.com:3456
HTTPS Proxy: http://proxy.domain.com:3456
No Proxy: localhost,127.0.0.1,172.172.0.0/16
[...]
```

Examples

This is an example of a /etc/systemd/system/docker.service.d/http-proxy.conf file that makes the Docker daemon use a proxy:

```
[Service]
Environment="HTTP_PROXY=http://user:password@proxy.domain.com:3456"
Environment="HTTPS_PROXY=http://user:password@proxy.domain.com:3456"
Environment="NO_PROXY=localhost,127.0.0.1,172.172.0.0/16"
```

This is an example of a ~/.docker/config.json that makes Docker propagate proxy configuration to the created containers:

```
{
        "proxies": {
            "default": {
                        "httpProxy": "http://user:password@proxy.domain.
com:3456",
                        "httpsProxy": "http://user:password@proxy.domain.
com:3456",
                        "noProxy": "localhost,127.0.0.1,172.172.0.0/16"
                }
        }
}
```

CPU and memory minimum requirements

The following table shows the minimum requirements for each service. Note that these are only minimum requirements. You should make sure to give each service enough resources depending on your system demands.

Service	Memory	CPU cores
wildflykiuwan-f[n]	2.5GB	2 cores
wildflykiuwan-a[n]	2.5GB per analysis slot	2 cores per analysis slot
wildflykiuwan-s[n]	2.5GB	2 cores
mysql	5GB	4 cores
loadbalancer	1GB	1 core
redis_0000[n]	2GB	2 cores

Note: CPU clock speed and disk speed will affect the overall response time.

With the above configuration, a system with the following load should give continuous service without problems:

- Parallel processing of 2 analyses (any additional parallel analysis request will be enqueued, and it will be executed as soon as any of the running analyses finishes)
- 50 concurrent web users or REST API calls.

Given the table above, for a **single-host installation** where no service is externalized the minimum system requirements are:

- 18GB of RAM and a processor with 8 cores for Kiuwan On-Premises.

It is recommended that you **overscale** these characteristics for the OS to have resources available for itself.

The Kiuwan On-Premises installation tool (kiuwan-cluster)

The Kiuwan On-Premises installation process is carried out by our "kiuwan-cluster" tool.

The tool is provided as a tar.gz file. The following table summarizes the resources you will find once the tool distribution is extracted:

Resource	Purpose
/config/volumes.properties	Configuration file to set where your persistent volumes will reside.
/docker/compose	Docker compose files used to manage Kiuwan On-Premises' docker services.
/docker/*.sh	Advanced shell scripts to interact with your Kiuwan On-Premises installation.
/logs	The folder where the tool will write installation logs.
/ssl	Tools that ease the certificate creation to keep Kiuwan On-Premises under a secure environment.
/support	Tools to ease collecting support data.
/tools	Internal tools used when installing.
/user-content	The folder where you will have to put some resources the installation process will need.
/volumes	The base persistent volumes (that may be copied to different locations depending on your installation needs).
*.sh	Main shell scripts to interact with your Kiuwan On-Premise installation.

The following sections will guide you through the installation process.

Installation: common steps

This guide will reference two important folders:

- **[INSTALLER_DIR]**: where the installation tool (kiuwan-cluster) will be located.
- **[VOLUMES_DIR]**: where the persistent volumes will be located.

Sometimes these folders will be referenced inside command line examples. Please make sure you replace any of them with the needed real path.

Note that it is up to you where these folders will be located.

Step 1: download kiuwan-cluster

The first step is to download kiuwan-cluster (the Kiuwan On-Premises installation tool). It can be downloaded directly from a terminal like this:

```
wget https://static.kiuwan.com/download/onpremise/kiuwan-cluster.tar.gz
```

This will download the latest available installation tool to the current directory.

Note that, as stated in System requirements, you will need access to static.kiuwan.com in order to download this file. You should also check your proxy configuration if you access the internet over a proxy server.

Step 2: untar kiuwan-cluster

Once downloaded, you should untar the provided gz file:

```
tar xvzpf kiuwan-cluster.tar.gz
```

This will untar the installation tool to a folder with extended version information of the tool. For example:

/home/user/kiuwan-cluster_master.XXXX-2.8.YYMM.V

This folder will be referred to as **[INSTALLER_DIR]** throughout this guide.

Step 3: copy license files

In order to be able to start a Kiuwan On-Premises installation, you will need two license files:

- configq1.zip
- license.zip

Copy these files to the user-content folder of your installation tool directory (please remember to replace [INSTALLER_DIR] with the real location of your installation directory):

```
cp configq1.zip [INSTALLER_DIR]/user-content
cp license.zip [INSTALLER_DIR]/user-content
```

Step 4: download and copy the needed driver version for MySQL

Kiuwan On-Premises needs this exact MySQL driver:

mysql-connector-java-5.1.39-bin.jar

You can download it by executing this command and extracting the jar file included inside the tar:

```
wget https://cdn.mysql.com/archives/mysql-connector-java-5.1/mysql-
connector-java-5.1.39.tar.gz
```

Untar the downloaded file:

```
tar xvzpf mysql-connector-java-5.1.39.tar.gz
```

Copy the connector jar file to the user content folder:

```
cp mysql-connector-java-5.1.39/mysql-connector-java-5.1.39-bin.jar
[INSTALLER_DIR]/user-content
```

Step 5: configure the volumes paths

The installation tool provides the base volumes needed to boot a first installation of Kiuwan On-Premises. Three volumes are included:

- config-shared: contains the base configuration, shared between different services.
- data-shared: contains the base data structure, shared between different services.
- data-local: contains the base data structure, independent for each service.

The installation tool needs to know where you want these volumes to reside. To do so, edit the file located in [INSTALLER_DIR]/config/volumes.properties and set desired locations:

```
config.shared=[VOLUMES_DIR]/config-shared
data.shared=[VOLUMES_DIR]/data-shared
data.local=[VOLUMES_DIR]/data-local
```

Please remember that [VOLUMES_DIR] is just a placeholder for the real path you chose.

Note that you will need to create the configured folders by running:

```
sudo mkdir [VOLUMES_DIR]
```

In case you are using different base directories for each volume, you must create all the needed base directories.

> Do NOT use the same folder for different volumes. It is MANDATORY that each volume resides on a separated folder.

Step 6: initialize your volumes

Copy the provided volumes to the configured location by running this script:

```
cd [INSTALLER_DIR]
sudo ./deploy-volumes.sh
```

Step 7: configure your email server

Kiuwan On-Premises needs a working and accessible e-mail server to send notifications.

Edit with your preferred editor the main configuration file, found in your [VOLUMES_DIR]:

```
sudo vim [VOLUMES_DIR]/config-shared/globalConfig/globalConfig.properties
```

> Please note that this is the file located in your [VOLUMES_DIR], not in the [INSTALLER_DIR], which only contains the base volumes.

Edit the following properties under the section named "Kiuwan instances shared configuration":

- kiuwan.mail.host: the host of your email server.
- kiuwan.mail.port: the port of your email server.
- kiuwan.mail.authentication: a flag to indicate if your mail server needs authentication or not.
- kiuwan.mail.username: the username to use when authenticating with your email server (only applies if kiuwan.mail.authentication is true).
- kiuwan.mail.password: the password to use when authenticating with your email server (only applies if kiuwan.mail.authentication is true).
- kiuwan.mail.from: the email account to use as the sender.
- kiuwan.mail.secure.layer: the security layer that your mail server uses (**ssl**, **tls** or **none**). Set to **none** if your mail server uses a plaintext connection.
- kiuwan.mail.secure.layer.value: the value to set to the ssl or tls (**true** or **false**). Only applies if kiuwan.mail.secure.layer is not none.
- kiuwan.default.mail.account: the email account to set to the default Kiuwan user (kiuwanadmin).

Mail server configuration examples

If your mail server uses a plaintext connection without authentication, set these properties values (other mail server properties values will be ignored):

- kiuwan.mail.authentication=false
- kiuwan.mail.secure.layer=none

If your mail server uses a TLS secure connection but does not need authentication:

- kiuwan.mail.authentication=false
- kiuwan.mail.secure.layer=tls
- kiuwan.mail.secure.layer.value=true

If your mail server uses a SSL secure connection and needs authentication:

- kiuwan.mail.authentication=true
- kiuwan.mail.username=myuser
- kiuwan.mail.password=mypassword
- kiuwan.mail.secure.layer=ssl
- kiuwan.mail.secure.layer.value=true

If your mail server uses a plaintext connection and needs authentication:

- kiuwan.mail.authentication=true
- kiuwan.mail.username=myuser
- kiuwan.mail.password=mypassword
- kiuwan.mail.secure.layer=none

Installation: single-host and minimum configuration

Follow this section if you want to proceed and install Kiuwan On-Premises with no further customization.

The defaults will install Kiuwan On-Premises with these characteristics:

- Single-host installation, including these services (see Kiuwan On-Premises Distributed System Architecture for more details):
  - Apache as a load balancer.
  - A Kiuwan front instance.
  - A Kiuwan analyzer instance.
  - A Kiuwan scheduler instance.
  - MySQL database.
  - Redis cluster.
- HTTPS support when accessing Kiuwan and between the loadbalancer and Kiuwan instances.
- Kiuwan On-Premises deployed in the default domain (https://kiuwan.onpremise.local).

If this is enough for you, just continue with the following steps.

If you plan to change the default domain, please refer to the Modifying the default domain section before continuing and come back here after you have made the needed changes.

Step 1: deploy user content

On a terminal, navigate to the [INSTALLER_DIR] folder and execute this command:

```
sudo ./deploy-user-content.sh
```

This will copy the user-content files to the configured volumes and set the needed permissions.

Step 2: install Kiuwan On-Premises

On a terminal, navigate to the [INSTALLER_DIR] folder and execute this command:

```
sudo ./install.sh
```

This will:

- Check if your host meets the minumum installation requirements.
- Download and run the needed Docker images.
- Install the database resources for Kiuwan On-Premises.
- Download the latest available Local Analyzer, Engine and Kiuwan for Developers to make them available in your installation.
- Install the engine data in your Kiuwan On-Premises database.
- Autogenerate the needed configuration for each Kiuwan instance.
- Run all the needed containers.

Once the installation is finisished please refer to the Accessing your Kiuwan On-Premises installation sect ion.

Accessing your Kiuwan On-Premises installation

In order to access your Kiuwan On-Premises installation you should follow a few more steps.

Step 1: add your domain to your local network DNS

To access your Kiuwan On-Premises installation you should take into account whether the selected domain is available in the DNSs your local network may use.

In order to access Kiuwan you will need to do one of the following options:

- Add kiuwan.onpremise.local to your DNS (recommended option).
- Add kiuwan.onpremise.local to your hosts file.

For testing purposes or if you choose the second option, edit this file in the host where you plan to access Kiuwan from:

- Windows OS: C:\Windows\System32\drivers\etc\hosts
- Linux OS: /etc/hosts

Add the following entry to the previous file:

```
[kiuwan_on_premise_host_ip] [kiuwan_on_premise_host]
```

For example, the previous entry may look like this for an installation pointing to the default host (note that the IP of the example may change in your local network):

```
192.168.0.56 kiuwan.onpremise.local
```

Step 2: add your certificates' CA to your clients

Depending on whether you are using a trusted CA or not to sign your certificates, you may need to add the CA to your client's certificate store to avoid warning messages**.**

Please refer to the Adding the provided or a custom CA to Kiuwan On-Premises' clients section for a complete explanation on how to handle this depending on your installation configuration.

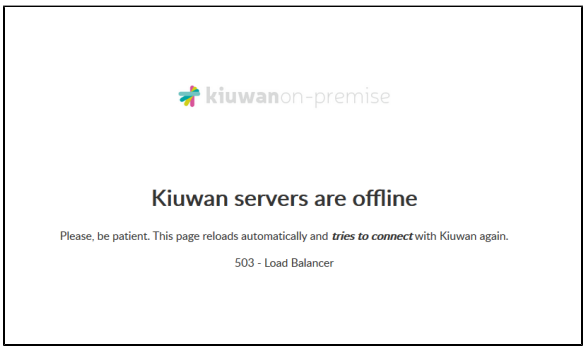
Step 3: access Kiuwan On-Premises

Accessing the web application

Once the previous steps have been done, you should be able to access Kiuwan On-Premises entering your Kiuwan host in your browser which by default is:

```
https://kiuwan.onpremise.local
```

Note that although the installation process may have finished, the Kiuwan servers may need some minutes to start up. If this is the case, a loading page will be shown (as long as you are using the provided Apache load balancer service):

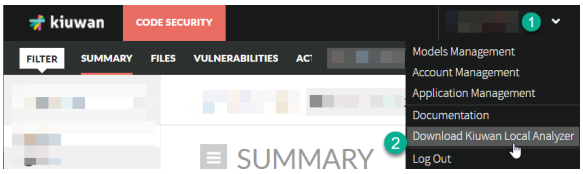


Once Kiuwan On-Premises services are started, you will be redirected to your Kiuwan On-Premises installation's main login page:



Downloading Kiuwan Local Analyzer

Once logged into the web application, you can download Kiuwan Local Analyzer by clicking on the "Download Kiuwan Local Analyzer" option in the top right drop-down menu.



Consuming Kiuwan REST API

To access your Kiuwan On-Premises installation via its REST API, you should point to this URL:

```
http(s)://[KIUWAN_DOMAIN]/saas/rest/v1
```

Please refere to the Kiuwan REST API documentation deployed in your Kiuwan On-Premises server for more details:

```
http(s)://[KIUWAN_DOMAIN]/pub/doc/rest-api/kiuwan-rest-api.html
```

You can also access the REST API documentation through the link shown in your Kiuwan On-Premises login page.

> There are no quota limits to Kiuwan REST API invocations anymore since Kiuwan On-Premises 2.8.1910.7.

Configuring Kiuwan for Developers

To install the Kiuwan for Developers plugin you should point to the corresponding download endpoint for each Kiuwan for Developers distribution:

IDE distribution	How to install	URL
Eclipse	Add a new updatesite	https://[KIUWAN_DOMAIN]/pub/updatesite
JetBrains	Add a new custom plugin repository	https://[KIUWAN_DOMAIN]/pub/jetbrains/plugins.xml
Visual Studio	Add an extension gallery	https://[KIUWAN_DOMAIN]/pub/vsgallery/atom.xml
Visual Studio Code	Download the extension package file and use the "Install from VSIX" option	https://[KIUWAN_DOMAIN]/pub/vscode/k4d-vscode.vsix

Please refer to the Kiuwan for Developers page for more information.

Default users

Kiuwan On-Premises is provided with two user accounts:

Username	Default password
sysadmin	sysadmin
kiuwanadmin	kiuwanadmin

Please make sure you change these passwords as soon as possible, by selecting the option "Account management" from the menu in the upper right corner and selecting "Change Password".