

HCL Clara V1.0.0.5 - Documentation

Table of Contents

1. Overview	5
About Clara documentation	5
Product Overview	6
What's New	8
Documentation in PDF format	11
Product Videos	12
Customer Support	13
2. Installation Guide	15
About the Installation Guide	15
System Requirements	16
Installing and Configuring Clara	17
Managing multiple Workload Automation environments	30
Upgrading Clara	33
Configuring Security	34
Customizing Clara Knowledge Base	36
Appendix	39
3. User's Guide	41
About the User's Guide	41
Managing User Credentials	42
Interacting with Clara	51
Clara replies to your questions	54
Clara runs actions on your behalf	56
Clara monitors your servers through HERO	70
Clara helps in problem determination	79
4. Troubleshooting Guide	87
About the Troubleshooting Guide	87
Collecting logs	88
Troubleshooting Clara	90

Overview

About Clara documentation

Welcome to the **HCL Clara** documentation, where you can find information about how to install, configure, and use HCL Clara.

Clara is an Intelligent Virtual Assistant that improves **Workload Automation** and **Workload Automation for Z** customer experience with natural language processing (NLP) interactions, enabling customized self-service automation with immediate 24x7 response as L1 and L2 assistant. Clara is the product expert that users can consult to learn the product capabilities. Clara goes beyond FAQs by executing routine tasks which increase end users satisfaction as the can get simple things done through contextual conversations.

For information about **Workload Automation and Workload Automation for Z**, see [HCL Workload Automation Documentation](#) (or [IBM Workload Automation Documentation](#) if you are using the IBM branded product).

Use the **Table of Contents** to navigate Clara documentation and find information about how to install, configure, and use the product.

[Product Overview](#)

[What's New](#)

[Documentation in PDF format](#)

[Product Videos](#)

[HCL Customer Support](#)

Product Overview

Workload Automation enables organizations to gain complete visibility and control over attended and unattended workloads. **HCL Clara**, an automation virtual assistant, understands human input, provides answers by tapping into a rich, specialized knowledge base, and allows direct interaction with Workload Automation in natural language to execute tasks. Clara saves IT operations time, frees up the schedulers, manages the how-to questions and initiates the initial troubleshooting of Workload Automation conveniently with voice command or simple text chat.

Capabilities

Enterprises depend on Workload Automation to manage critical business processes, while reducing operating costs. IT Operators ensure Workload Automation operation continuity 24x7, avoiding involvement of support team. Application support teams want to quickly solve issues in their applications to ensure business continuity. Clara makes Workload Automation even easier to use, by offering a conversational interface to scheduling users, access the company's automation knowledge base, providing recommendations and performing tasks for the end-users.

Clara provides the following capabilities:

- Interactions in natural language
- Easy and intuitive training modules
- Intuitive conversation definition via UI
- Action script and external language support for integration
- External interface for third party integration
- Robust AI/NLP based intent identification engine
- Voice recognition
- Pluggable/customizable knowledge base
 - HCL knowledge base with regular updates
 - Possibility to customize the knowledge base upon customer needs and to extend the knowledge base from Clara UI
- Actions and Runbooks implementation leveraging product APIs
 - Clara leverages HCL software products APIs to enable users to automate business processes and troubleshoot problems

Benefits

Clara offers a human-like, personalized, round-the-clock experience to Workload Automation users, which will:

- Minimize the FAQ-type calls
- Speed up learning curve on new features
- Reduce common-issue and repetitive incidents
- Promote best practices

Simple to get started

Learn how to use various functions and workarounds, and get your actions done on Workload Automation in simple conversations at your convenience with Clara.

Manage how-to questions

- Get me started with building my process (jobs, APIs...)
- How to use calendar or event-based scheduling
- Understand new features

Support basic scheduling scenarios

- Submit a job or job stream
- Re-run jobs

Troubleshoot Workloads

- Why the job is not starting?
- Why the job failed?

What's New

See what's new with Clara releases.

Clara V1.0.0.5

Clara V1.0.0.5 includes the following enhancements:

1. Support of WA V9.5 FP4.
2. A new optional parameter - THREADPULL_SIZE - has been added to the Workload Automation environment configuration file to specify the number of monitoring processes that Clara can manage simultaneously. Default value is set to 50.

Satisfied Requests for Enhancements (RFEs):

1. Clara can provide information about Workload Automation error messages. Ask Clara "Search <error_message_code>, or "Search error <error_message_code>" to get an explanation of the error message you've received.
2. Full support of user name parsing in Clara's email.
3. Clara replies can contain special characters.
4. Clara can send emails also to SMTP servers that are not authenticated.
5. Clara's Credential Manager supports user names containing special characters or spaces.

Fixed issues:

1. When a user asks to open a ticket, Clara provides the link to the Service Platform specified in the Workload Automation environment configuration file. However, in Clara's reply, the text always refers to ServiceNow.
2. Wrong Clara's reply when user asks how to manage credentials.
3. Wrong Clara's reply to user input such as "Hi Clara" or "Hello Clara".
4. The description of the "Submit a job" function has been corrected inside the "What can you do?" menu.
5. './clara.sh --load-project' was not working properly for port values different from 443.
6. Wrong links to Automation Hub have been fixed in Clara's replies.
7. The logout button erroneously reloads login page within the chat frame.
8. The action "Submit a job" doesn't work when the workstation belongs to a folder.

9. RabbitMQ Dockerfile issues have been fixed for Clara online installation.

Clara V1.0.0.4

Clara V1.0.0.4 includes the following enhancements:

Availability of Clara for WAz

New set of Q&As for WA V9.5 FP3 and Automation Hub

Submit a service from Self-Service Catalog

Clara can submit a service from Self-service Catalog on your behalf, with no need to use a VPN. See [Submit a service from Self-service Catalog](#).

Notifications

Clara can notify you about a Job or Service status (started, completed successfully, completed with errors, in error, in late). See [Monitor a job](#).

Share information

You can share information provided by Clara with other users, via email. See [Copy and share the status of a job](#).

Clara can also assist opening a ticket to the Service Platform of your choice. See [Open a ticket](#).

Server monitoring through HERO

Clara can monitor and get information about your WA servers on your behalf, through HERO (HEalthcheck & Runbook Optimizer). This feature is available for WA users only (not WAz). See [Clara monitors your servers through HERO](#).

Support of Slack and Sametime messaging tools

You can contact Clara also from Slack and Sametime.

Clara V1.0.0.3

Clara V1.0.0.3 includes the following enhancements:

Push notifications

Download information

Delayed actions

Documentation in PDF format



HCL Clara documentation is available in **PDF** format:

Product Videos

You can find a detailed explanation of the capabilities and benefits provided by **Clara for Workload Automation** in videos demonstrating user scenarios:

[Clara for Workload Automation](#)

[Clara for Workload Automation for Z](#)

[HCL Workload Automation - Automate more, automate better!](#)

[Automate more, better, and smarter with HCL Automation Power Suite](#)

[How to install HCL Clara V1.0.0.3](#)

Customer Support

To contact HCL Customer Support or create a product case, see: [Customer Portal](#).

Installation Guide

About the Installation Guide

The Installation Guide provides information about how to install and configure HCL Clara.

[System Requirements](#)

[Installing and Configuring Clara](#)

[Managing multiple Workload Automation environments](#)

[Upgrading Clara](#)

[Configuring Security](#)

[Customizing Clara Knowledge Base](#)

[Appendix](#)

System Requirements

Clara is installed using Docker.

You can install Clara on Linux 64 bit operating system.

Supported browsers

- Google Chrome 67.0.3396.99 or higher
- Mozilla Firefox 61.0.1 or higher
- Microsoft Edge 79 or higher

Software requirements

- Docker 17.12.0+
- Docker Compose 1.24.0+
- Red Hat Linux 7.5+
- CentOS 7.5-1804+

Hardware requirements

Clara must be installed and configured with the following minimum requirements:

- CPU 64 bit, 4+ core
- RAM 8GB+ (recommended 16GB)
- Storage 100GB HDD

Workload Automation requirements

Clara can assist Workload Automation users starting from **WA version 9.3 fix pack 4**.

Installing and Configuring Clara

Checking system prerequisites

Before starting to install Clara, you must check the following system prerequisites:

1. Verify that Docker and Docker Compose are installed, configured, and ready to use. For the required version, see: [System Requirements](#).

If you don't have Docker and Docker Compose already installed, see: [Installing Docker and Docker Compose](#).

2. Clara requires some values to be set for **ulimit** parameter, for Linux OS. See: [How to verify and set ulimit parameter](#).
3. Verify the available virtual memory. See: [How to verify and set the available virtual memory](#).
4. If you are installing Clara on RHEL or CentOS distros, SELinux must be set to Permissive or Disabled. See: [How to set SELinux to permissive](#).

Installation procedure

To install and configure Clara, run the following procedure.

1. From [HCL License Portal](#) download the appropriate Clara installation package. Depending on your license type, you have two different options:
 - HCL CLARA for Workload Automation Assistant Standard Level v1.0.0.4
 - HCL CLARA for Workload Automation Assistant Entry Level v1.0.0.4

Each option is available with 2 different types of installation packages: online and offline.

HCL SOFTWARE

Home | Activation & Entitlements | License Support | Devices | Downloads | Accounts & Users

Downloads

HCL CLARA For Workload Automation Assistant Entry Level v1.0.0.4

Files | Download Log | Notification Log

Download Selected Files

<input type="checkbox"/>	+	File Description	File Size	File Added	File Name
<input type="checkbox"/>	+	HCL CLARA for Workload Automation v1.0.0.4	587.89MB	Dec 18, 2020	HCL_CLARA_1.0.0.4_ONLINE.tar.gz
<input type="checkbox"/>	+	HCL CLARA for Workload Automation v1.0.0.4 - OFFLINE INSTALLATION	2.15GB	Dec 18, 2020	HCL_CLARA_1.0.0.4_OFFLINE.tar.gz

Use the **online installation package** to build Clara images directly on your machines. This type of package requires a working Internet connection.

Use the **offline installation package** if the machine does not have an Internet connection or if you do not want to build Clara.

The following table summarizes the content of the 2 different types of packages:

HCL_CLARA_1.0.0.4_ONLINE.tar.gz	HCL_CLARA_1.0.0.4_OFFLINE.tar.gz	
Dimension around 588 MB	Dimension around 2.15GB	
Require internet connection to build Clara images	Images are available in tarball. Internet not required	
Contents of package		Description
HCL_CLARA/	HCL_CLARA/	
clara.sh	clara.sh	Installation script
clara.tar.gz	clara.tar.gz	Clara docker-compose files
clara-img.tar.gz	-	Source files to build Clara images
	clara_images.list	List of Clara images
	clara_images.tar.gz	Clara images that can be loaded using "./clara.sh --load-images"
	project-wa.list	List of projects images (only one for now)
	project-wa.tar.gz	Clara image with KB for Work. Automation. This image can be loaded using "./clara.sh --load-images"

2. Extract the content of the tar.gz file into <BUILD_DIR>, a directory of your choice using one of the extraction tools available on your system or downloadable from the Internet. The tool you use must be able to keep the file permissions on the extracted files.
3. To enable Clara interaction with your Workload Automation environment, you must properly customize the <BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE file. See below the section about [Customizing environment files](#). If you want to enable Clara interaction with multiple Workload Automation environments, see [Managing multiple Workload Automation environments](#). You might decide to skip this step: in this case, the installation script will ask you to provide the required customization information during the installation process.
4. If you want to change default Clara credentials before building in Docker, see [Customizing Clara credentials](#).
5. To install Clara on Linux operating system, the user must have read and write permissions for the <BUILD_DIR> directory. Also, the user must have execute permission for Docker commands. This means that the user must be a member of sudoers group or Docker group. If the user is a member of sudoers group but not of Docker group, the installation script must be run with sudo.
6. Open a bash shell and get ready to install Clara.
7. If it is the **first time** you install Clara, or you need to **update** Clara images, you must first build Clara images. You can skip this step if you are not installing Clara for the first time or the images are locally available.

Change the directory to <BUILD_DIR> and extract the content of clara-img.tar.gz file:

```
cd <BUILD_DIR>
tar xvfz clara-img.tar.gz
```

Build Clara images by running the command:

```
clara-img/clara-build.sh -p -i ALL 2>&1 | tee clara-build.log
```

Note: To run the above command, an active internet connection is required. In case this is not possible, you can build Clara images on a different Linux server with Docker and an internet connection. See [Building Clara images on a different server](#).

If you have built images locally, you can skip directly to the next step.

If you have built images on a remote server, ensure all required tar.gz files have been copied locally under <BUILD_DIR> directory and run the following command:

```
./clara.sh --load-images
```

8. Proceed with Clara installation. See [Clara.sh command](#) usage if you want to skip license check, OS prerequisites check, all variable prompts and use defaults (defaults will be loaded from .env file if present in the installation or local path). For example, if you pass the following parameters to the installation script:

```
--skip-check-prereq --skip-var-prompts
```

and a valid .env file if present in the installation or local path, the installation will be **completely unattended**. Otherwise:

- **If the Autobot License File is not yet available**, run the following steps:
 - a. From the <BUILD_DIR> directory, run the command:


```
./clara.sh --install --install-path <install_path>
```

 where <install_path> is the installation directory of your choice (default value is /opt/hcl).
 For a detailed description of the command, see [clara.sh command](#).
 If Clara is successfully installed, the installation process generates a UID.
 - b. Contact HCL Solutions Team (HCLSolutionsSupport@hcl.com) with the UID to get a valid Autobot license (license.properties file).
 - c. Copy the **license.properties** file in the Clara installation directory.
 - d. From the installation directory, activate and configure Clara by running the following command:


```
./clara.sh --load-project <prj> --license license.properties
```

 where <prj> can be:
 - IWA** IBM Workload Automation
 - HWA** HCL Workload Automation
 - IWAz** IBM Workload Automation for z
 - HWAz** HCL Workload Automation for z

- **If you already have a valid Autobot License File**, run the following steps:
 - a. From the <BUILD_DIR> directory, install Clara by running the following command:


```
./clara.sh --install --install-path <install_path> --license ./license.properties
```
 - b. From the installation directory, activate and configure Clara by running the following command:


```
./clara.sh --load-project <prj>
```

- **If you are reinstalling Clara and the persistent Clara volume is still available**, you can reuse the license already cached in the volume. Run the following steps:
 - a. From the <BUILD_DIR> directory, install Clara by running the following command:


```
./clara.sh --install --install-path <install_path> --license CONTAINER
```
 - b. From the installation directory, activate and configure Clara by running the following command:


```
./clara.sh --load-project <prj>
```

9. As the installation script starts, if you haven't previously customized the `<BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE` file (see step 3), you are required to provide information to enable Clara interaction with your target environment. Press **Enter** to accept the proposed default values (see step 3).
10. The installation script checks the prerequisites (unless you have specified the `--skip-check-prereq` parameter), runs the installation process, and verifies its successful completion.

Post installation steps

When the installation is completed, you can access Clara web interface at the following link <https://<IP:PORT>/Clara/> where:

- **IP** is the IP address of Clara machine
- **PORT** is the port of Clara machine. You can skip it if you use the default port (443)

The installation script generates two Clara users:

- userid **solutions**, password **Hclsolutions00**, with user role
- userid **admin**, password **Hclsolutions00**, with administrator role

As you start chatting with Clara, you must define credentials for the product environments you want to manage with Clara. The User Credential Manager component will help you map your product environment credentials with your Clara credentials (each Clara user can manage his own Workload Automation credentials). For additional details, see [Managing User Credentials](#).

Use the Keycloak administration console to define new Clara users, new roles, or change default passwords. You can access Keycloak administration console <https://<IP:PORT>/keycloak/auth/admin> by using the following default credentials:

- userid=**admin**
- password=**Hclsolutions00**

If you want, you can change Keycloak default password. For details about using Keycloak, see [Configuring Security](#).

For security reasons, you are recommended to generate a custom Keycloak Client Secret, in place of the default one provided in your environment TEMPLATE file. See [How to set a custom Client Secret](#).

After installing Clara, to enable Clara interaction with multiple Workload Automation environments, see: [Managing multiple Workload Automation environments](#).

You can also interact with Clara by using messaging tools such as **Slack** and **Sametime**. For the required configuration steps, see [Installing Adapters for Messaging Tools](#).

To start/stop Clara containers and to start/stop Clara services, see [clara.sh command](#) options below.

Customizing environment files

Both for Workload Automation and Workload Automation for z, customize the <BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE file by providing the following information:

NGINX_PORT	TCP/IP Port to use with Clara [Default: 443]
NGINX_CERTS	Path where nginx certificates for Clara are stored [Default: ./nginx]
CLIENT_SECRET	Client Secret to access Keycloak [Default: 214c76c5-f726-482b-92b6-2f5586c7bbe1]
PROJECT	HWA for HCL Workload Automation IWA for IBM Workload Automation HWAz for HCL Workload Automation for z IWAz for IBM Workload Automation for z
WA_ENV_NAME	Mnemonic name of the Workload Automation environment [Default: WA-PROD]
WA_SERVER_NAME	Only for WA: hostname of the Workload Automation server (MDM) [Default: wa-server]. You can skip this settings if you are configuring Workload Automation for z
WA_SERVER_PORT	Only for WA: HTTPS port where WA server (MDM) is listening [Default: 31116]. You can skip this settings if you are configuring Workload Automation for z
WA_CONSOLE_NAME	Hostname of the Workload Automation Web console [Default: wa-console]
WA_CONSOLE_PORT	HTTPS port of the Workload Automation Web console [Default: 16311]
WA_ENGINE_NAME	Name of the engine definition within Workload Automation [Default: engine]
WA_LIC_SERVER_NAME	Hostname of the License Service provided by HCL [Default: https://flex1513.compliance.flexnetoperations.com/instances/]
WA_LIC_ID	Unique ID (License Server ID) of the device your Clara license entitlement has been associated with
WA_TKT_URL	Url to the Service Platform for Ticketing [Default: https://servicenow.com]
THREADPULL_SIZE	The number of monitoring processes that Clara can manage simultaneously [Default: 50] - Optional parameter
SMTP_HOST	Fully qualified hostname of the SMTP Server that will be used by Clara to send email notifications [Example: smtp.gmail.com]
SMTP_SENDER	Used to send email to the recipient specified in Clara Credential Manager [Example: username@gmail.com]
SMTP_SENDER_PASSWORD	Used to authenticate the SMTP_SENDER
SMTP_AUTH_PROTOCOL	The authentication type defined in the SMTP_HOST server. Can be TLS or SSL [Default: TLS]
SMTP_PORT	The port of the SMTP mail server [Example: 465 for gmail]
HERO_IP	Needed for HERO integration. HERO server IP address

HERO_PORT	Needed for HERO integration. HERO server Port
HERO_SECRET	HERO client secret to connect: "CLIENT-SECRET-CONFIG"
HERO_CLIENT_ID	HERO nginx client ID: "nginx"
HERO_REALM	HERO realm info for the connection: "hero"
HERO_URL	Landing page of HERO if not installed: https://www.hcltechsw.com/products/hero/offerings/hero-workload-automation
IMG_PREFIX	Clara docker image repo prefix when images are not available locally, like: example.com/
TAG	Clara docker image tag based on Clara version, optional [Default: latest]

If a re-configuration is done on dynamic files, such as settings.properties or nginx.conf, restart docker-compose by running the following commands:

- **docker-compose stop**
- **docker-compose start**

or, you can restart just the involved container, for example **clara-autobot** for settings.properties, or **clara-nginx** for nginx.conf and certificates.

If the configuration is on the client, try to delete the cache on browser.

If the configuration is on the docker-compose.yml, then rebuild the entire application by running the following command:

docker-compose up -d --build

Building Clara images on a different server

To build Clara images on a different server, run the following steps:

1. Move clara-img.tar.gz file, available under <BUILD_DIR> directory, on a Linux server with Docker and an active Internet connection.

2. Extract the content of the file clara-img.tar.gz

tar xvfz clara-img.tar.gz

3. Build Clara images running the command:

clara-img/clara-build.sh -p -i ALL 2>&1 | tee clara-build.log

4. Create the archive file with all required docker images that can be imported later.

To build the archive, run the following command:

clara-img/clara-build.sh -s 2>&1 | tee clara-save.log

5. The following files will be generated:

- clara_images.tar.gz Clara images

- `clara_images.list` Clara images list (to build correct names)
 - `project-iwa.tar.gz` IBM Workload Automation Clara project
 - or
 - `project-iwaz.tar.gz` IBM Workload Automation for z Clara project
 - `project-hwa.tar.gz` HCL Workload Automation Clara project
 - or
 - `project-hwaz.tar.gz` HCL Workload Automation for z Clara project
6. Move the above files under the directory `<BUILD_DIR>` on the server where Clara will be installed
 7. Remove the `clara-img` directory, the `clara-img.tar.gz` file and any file produced. You can also remove the images from docker using the command `"docker rmi <name>"`.

Customizing Clara credentials

If you want to change default Clara credentials before building in Docker, you can edit them inside the following files:

1. Default **admin** credentials can be changed inside the **"docker-compose.yml"** file, in the root docker installation folder.
 - The username can be changed by editing the field **KEYCLOAK_USER** under the keycloak container section.
 - The password can be changed by editing the field **KEYCLOAK_PASSWORD**
2. Default **test** credentials inside Clara Realm can be edited in the **./keycloak/Dockerfile** where you find a reference to `add-user-keycloak.sh` file, customizing the following two parameters:
 - **--user** for the user
 - **--password** for the password

Installing Adapters for Messaging Tools

You can interact with Clara by using **Slack** and **Sametime** messaging tools.

To install the required adapters, run the following procedure:

From the `<BUILD_DIR>` directory where you extracted Clara installation package, build the adapter image for the adapter that you want to install. Run the command:

- For Slack: `clara-img/clara-build.sh -p -i sIAdapter`
- For Sametime: `clara-img/clara-build.sh -p -i stAdapter`

If you want to install the adapter on the same machine where Clara is installed, run the following steps:

1. Properly configure the **config.properties** file containing the adapter properties, located in the following directory:

- For Slack: **<BUILD_DIR>/Adapters/SlackAdapter/resources**. For details see: [Configuration file for Slack adapter](#)
 - For Sametime: **<BUILD_DIR>/Adapters/SametimeAdapter/resources**. For details see: [Configuration file for Sametime adapter](#)
2. Start the adapter:
 - For Slack: from **<BUILD_DIR>/Adapters/SlackAdapter**, run the command:
docker run --env-file resources/config.properties --name clara-slackadapter -d clara-slackadapter:latest
 - For Sametime: from **<BUILD_DIR>/Adapters/SametimeAdapter**, run the command:
docker run --env-file resources/config.properties --name clara-sametimeadapter -d clara-sametimeadapter:latest

If you want to **install the adapter on a different machine**, run the following steps:

1. Save the adapter image to a file by running the following command:
 - For Slack: **docker save clara-slackadapter:latest | gzip > clara-slackadapter.tar.gz**
 - For Sametime: **docker save clara-sametimeadapter:latest | gzip > clara-sametimeadapter.tar.gz**
2. Locate the configuration file containing the adapter properties:
 - For Slack: **<BUILD_DIR>/Adapters/SlackAdapter/resources/config.properties**
 - For Sametime: **<BUILD_DIR>/Adapters/SametimeAdapter/resources/config.properties**
3. Copy the **tar.gz** file containing the adapter image and the configuration file to the machine where you want to install the adapter.
4. Properly configure the **config.properties** file. For details see:
 - For Slack: [Configuration file for Slack adapter](#)
 - For Sametime: [Configuration file for Sametime adapter](#)
5. From the machine where you want to install the adapter, load the image from the file by running the following command:
 - For Slack: **docker load --input clara-slackadapter.tar.gz**
 - For Sametime: **docker load --input clara-sametimeadapter.tar.gz**
6. From the folder where you loaded the adapter image, start the adapter by running the following command:
 - For Slack: **docker run --env-file <PATH>/config.properties --name clara-slackadapter -d clara-slackadapter:latest**
 - For Sametime: **docker run --env-file <PATH>/config.properties --name clara-sametimeadapter -d clara-sametimeadapter:latest**

where **<PATH>** is the folder containing the config.properties file

Configuration file for Slack adapter

auth_token	The token generated by Slack bot
client_secret	The client secret requested to access Keycloak, as defined in the .env environment configuration file for Clara
bot_url	In the format <IP>:<PORT>. IP address and Port of the machine where Clara is installed (if Port is 443, it can be omitted)
bot_project	Specify WA if the Clara project specified during the installation is HWA or IWA. Specify WAZ if Clara project is HWAZ or IWAZ.
timeout_interval	The time (in milliseconds) that the Slack session should stay active while chatting with Clara (default: 200000)

Configuration file for Sametime adapter

sametime_password	Password for Clara Sametime account
sametime_username	Username for Clara Sametime account
is_encrypted=false	Encryption flag for Sametime account password. Initial default: false. After the first connection the password is automatically encrypted and the flag will turn on true
sametime_community_hostname	Sametime community server host name
bot_host	In the format <IP>:<PORT>. IP address and Port of the machine where Clara is installed (if Port is 443, it can be omitted)
client_secret	The client secret requested to access Keycloak, as defined in the .env environment configuration file for Clara
bot_project	Specify WA if the Clara project specified during the installation is HWA or IWA. Specify WAZ if Clara project is HWAZ or IWAZ
bot_language=en	The Sametime language
is_secured=true	Set it to true for HTTPS connection, false for HTTP
auth=true	Keycloak enablement flag. Set to false if you want to disable Keycloak.

clara.sh command

clara.sh [options]

OPTIONS:

Help:

-h Help

To control Clara services:

--up [up options] Create and start Clara container

where [up options] can be:

 [--nc] Specify --nc to get the new configuration from the .env file

 [--noauth] Specify --noauth to disable authentication

 [--auth] Specify --auth to enable authentication

--down [--vol] Stop and remove Clara containers and network

 Specify --vol to remove also persistent volumes

--start Start Clara services

--stop Stop Clara services

--join-net Link Clara network to Workload Automation network when they are both running in the same network

--remove Print commands to clean-up Clara containers, persistent volumes and network manually

--print-urls Print Clara URLs

Examples:

First Clara start-up

./clara.sh --up

Stop Clara containers

./clara.sh --stop

To add a custom Knowledge Base to Clara:

`--kb <CSV_file>` Integrate Clara KB with the provided Q&A
`<CSV_file>` The CSV file with the KB (use absolute path)

To backup and restore Clara persistent volumes:

`--backup-volumes` Create a backup file for each Clara persistent volume
`--restore <vol_name> --from-dir <backup_directory>`
Restore the volume `<vol_name>` from the backup stored in
the directory `<backup_directory>` (use absolute path)

Examples:

Backup all clara volumes

`clara.sh --backup-volumes`

Restore the clara-Etc_data volume taken on 2020-12-09-1439-734236800

`clara.sh --restore clara-Etc_data --from-dir /opt/hcl/clara/backups/2020-12-09-1439-734236800`

To install Clara:

`-c, --prereq` Check Prerequisites (skip any other option if present)

`--load-images` Load Clara images from clara_images.tar.gz(in the same path as this script)

`--install` Install or upgrade Clara

`--install-path <path>` Modify install path (default is /opt/hcl)

`--load-project <prj>` Install the specified Clara project

Possible values for `<prj>` are:

IWA IBM Workload Automation

HWA HCL Workload Automation

	IWAz	IBM Workload Automation for z
	HWAz	HCL Workload Automation for z
<code>--license <lic_file></code>	Import Clara license	<p><code><lic_file></code> fully qualified path to the license file</p> <p>Use CONTAINER if Clara was already enabled in the past and the persistent storage is available</p>
<code>--migrate-data</code>	Migrate data from Clara V1.0.0.2 or V1.0.0.3 to V1.0.0.4 or above	
<code>--skip-lic-check</code>	Pass this parameter to skip license check	
<code>--skip-check-prereq</code>	Pass this parameter to skip all OS prereq checks	
<code>--skip-var-prompts</code>	Pass this parameter to skip all var prompts except prereq checks and use defaults	Defaults will be loaded from <code>.env</code> file if present in the installation or local path.
<code>--remote-hostname <name></code>	Specify Clara hostname, i.e. can be the hostname for Clara external access	
Steps:		

1. Check Clara prerequisites

```
./clara.sh -c
```

2. Load Clara images (run only if you need to import new images)

```
./clara.sh --load-images
```

3. Install or upgrade Clara. Select one of the following cases:

a. First time installation (if you don't have a valid `license.properties` file), run:

```
./clara.sh --install --install-path /opt/hcl
```

Communicate the generated UID to the HCL Solutions Team (HCLSolutionsSupport@hcl.com) to get a valid `license.properties` file.

b. If you have a valid `license.properties` file, run:

```
./clara.sh --install --install-path /opt/hcl --license ./license.properties
```

c. If Clara is already installed and activated and the persistent Clara volume is available, you can re-use the license already cached. Run:

```
./clara.sh --install --install-path /opt/hcl --license CONTAINER
```

4. Configure Clara for Workload Automation. From the installation directory, run:

```
cd /opt/hcl/clara  
./clara.sh --load-project HWA
```

Note: This step will overwrite the existing Clara Knowledge Base and Smart Actions. If you previously modified the Knowledge Base and don't want to lose your changes, see: [Upgrading Clara](#).

5. If you are upgrading Clara from 1.0.0.2 or 1.0.0.3 to V1.0.0.4 or above, run:

```
cd /opt/hcl/clara  
./clara.sh --migrate-data
```

For unattended installations, add the following options to the above commands:
--skip-check-prereq --skip-var-prompts

Managing multiple Workload Automation environments

During the installation process, to enable Clara interaction with multiple Workload Automation environments, you must properly configure the **CLARACONFIGWA** keyword in the <BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE file. For Workload Automation for z, configure the **CLARACONFIGWAZ** keyword.

This keyword contains a json object (**envs**) with informations about the different Workload Automation environments. See examples below.

After Clara installation, to add additional Workload Automation environments you must modify the .env file defined during the installation. After you have modified the Clara-WA.env, run the following command to activate the updates:

```
clara.sh --up --nc
```

In Example 1, **envs** contains only one environment named WA-PROD.

Example 1

For Workload Automation:

```
CLARACONFIGWA = '{"envs": { "WA-PROD": { "ip": "mdm-host.example.com", "port": "31116", "engineId": "engine_wa-server", "dwcGrp": "DWC-1"}, "dwcEnvsGrp": { "DWC-1": { "dwclp": "dwc-host.example.com", "dwcPort": "16311"}, "license": { "licenseServer": "https://hclsoftware.compliance.flexnetoperations.com/instances/", "licenseID": "38K36F2YXS18"}, "tkt_url": "https://servicenow.com", "product": "WA"}}
```

For Workload Automation for z:

```
CLARACONFIGWAZ = '{"envs": { "WA-PROD": { "engineId": "engine_wa-server", "dwcGrp": "DWC-1"}, "dwcEnvsGrp": { "DWC-1": { "dwclp": "dwc-host.example.com", "dwcPort": "16311"}, "license": { "licenseServer": "https://hclsoftware.compliance.flexnetoperations.com/instances/", "licenseID": "38K36F2YXS18"}, "tkt_url": "https://servicenow.com", "product": "WAZ"}}
```

If you want to add another environment, you must add another element with all the requested information. In Example 2, **envs** contains two environments named Bot1 and Bot2.

Example 2

For Workload Automation:

```
CLARACONFIGWA = '{"envs": { "Bot1": { "ip": "54.89.190.168", "port": "", "engineId": "engine_waadmin", "dwcGrp": "dwcGroup1"}, "Bot2": { "ip": "192.168.1.15", "port": "", "engineId": "engine_waadmin"}, "dwcEnvsGrp": { "dwcGroup1": { "dwclp": "localhost", "dwcPort": ""}, "product": "WA", "license": { "licenseServer": "https://flex1513-uat.compliance.flexnetoperations.com/instances/", "licenseID": "38K36F00XS18"}, "tkt_url": "https://servicenow.com"}}
```

For each environment, you specify the following information:

- **env_name**, the name of the Workload Automation environment. It is used as an index for the **envs** json objects. In the above example, “Bot1” and “Bot2” are the environment names.
- **ip**, the hostname of the Workload Automation engine (not needed for Workload Automation for z)
- **port**, the port to connect the Workload Automation engine (not needed for Workload Automation for z)
- **engineId**, the name of the engine definition within Workload Automation
- **dwcGrp**, the name of the group which contains information about the connection with the Dynamic Workload Console. If not specified, the environment will get the connection inside the “default” group.

In addition to the **envs** json object, the **dwcGrp** json object can also be configured.

The **dwcGrp** json object defines one or more groups to set the connection to the **Dynamic Workload Console**. For each group, you must specify the following information:

- **groupName**, the name of the group. It must match the **dwcGrp** property of the **envs** object
- **dwcIp**, the hostname of the Dynamic Workload Console
- **dwcPort**, the port of the Dynamic Workload Console

Moreover, you specify:

- **product**, the ID of the product managed by Clara (WA or WAz)
- **licenseServer**, the hostname of the License Service (Default: <https://flex1513.compliance.flexnetoperations.com/instances/>)
- **licenseID**, the unique ID of your Clara license
- **tk_url**, the Url to Service Platform for Ticketing
- **threadpull_size**, the number of monitoring processes that Clara can manage simultaneously (Default: 50) - Optional parameter

In the **dwcGRP** json, if you set “**default**” as **groupName**, all the **envs** that do not have **dwcGrp** specified will get the default connection. See the following example.

Example 3

For Workload Automation:

```
CLARACONFIGWA = '{"envs": { "Bot": { "ip": "localhost", "port": "", "engineId": "engine_waadmin"}}, "dwcEnvsGrp": {  
  "default": { "dwcIp": "localhost", "dwcPort": ""}, "product": "WA", "license": { "licenseServer": "https://flex1513-  
uat.compliance.flexnetoperations.com/instances/" , "licenseID": "38K36F11XS18"}, "tk_url": "https://servicenow.com"}
```

Configuring Clara integration with HERO

You can take advantage of **CLARACONFIGHERO** keyword in the <BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE file to configure Clara integration with HERO.

See the following example:

```
CLARACONFIGHERO='{ "envs": { "HERO": { "ip": "10.14.37.117", "port": "8080", "hero_secret": "CLIENT-SECRET-CONFIG", "client_id": "nginx", "realm": "hero" }, "license": { "licenseServer": "https://hclsoftware.compliance.flexnetoperations.com/instances/" , "licenseID" : "38K36F2YXS19"}, "tkr_url": "https://servicenow.com", "product": "HERO" } }
```

For each HERO environment, you must specify the following information:

- **env_name**, the name of the HERO environment
- **ip**, the IP address of the HERO server
- **port**, the port of the HERO server
- **hero_secret**, Hero client secret
- **client_id**, HERO nginx client id
- **realm**, HERO realm info for connection
- **licenseServer**, the hostname of the License Service (Default: <https://flex1513.compliance.flexnetoperations.com/instances/>)
- **licenseID**, the unique ID of your Clara license
- **tkr_url**, the Url to Service Platform for Ticketing
- **product**, HERO

Upgrading Clara

You can upgrade your Clara installation from V1.0.0.2 or V1.0.0.3 to V1.0.0.4 or above.

1. To upgrade Clara to V1.0.0.4 or above, follow the [Installation Procedure](#) from step 1 to step 8. For step 8, follow the branch "*If you are reinstalling Clara and the persistent Clara volume is still available*".

Note:

When you run the command `./clara.sh --load-project <prj>` to activate and configure Clara, the installation script will add FAQs for the new version to the existing Clara's Knowledge Base, overwriting FAQs with the same ID. Existing FAQs with unique ID will be maintained.

2. To migrate the users defined with the Credential Manager in your previous installation, run the following command from the installation directory:

```
./clara.sh --migrate-data
```

Integrating Clara Knowledge Base

Before upgrading Clara, if you have modified the Knowledge Base of your previous installation by adding custom FAQs, you might need to export and edit them:

1. From Clara Administration Console (https://<Clara_machine_IP>/botAdministrator/index.html), export the FAQs into an Excel file by clicking the **Export** button.
2. Delete all the FAQs except your custom ones and save the file.

After the upgrade, to import an Excel file (for example ACME-KB.csv file) containing your custom FAQs, run the command:

```
./clara.sh --kb /opt/hcl/ACME-KB.csv
```

Configuring Security

In Clara, user authentication is managed by Keycloak.

In Keycloak, each application has its own Realm with different users and authorization settings. Clara authorization settings are stored in a Realm named Clara.

For details about Keycloak, see [Keycloak documentation](#).

The steps to configure the security for your Clara installation, including the generation of a new secret, and the customization of SSL certificates, are run automatically by the installation script.

The installation script generates two Clara users:

- userid **solutions**, password **Hclsolutions00**, with user role
- userid **admin**, password **Hclsolutions00**, with administrator role

To add additional users, roles, or to change the default passwords, see the steps in **Creating a new user** below.

Creating a new user

The installation process generates a Keycloak default realm named **Clara** and a default client named **Clara**.

For additional information about Keycloak realms and clients, see [Keycloak documentation](#).

Use the Keycloak administration console to define new users, new roles, or change user passwords.

For example, to create a new Clara user with administrator role, run the following steps:

1. Access Keycloak administration console **https://<IP:PORT>/keycloak/auth/admin** by using the following credentials:
 - userid=**admin**
 - password=**Hclsolutions00**
2. If you want, you can change Keycloak default password:
 - a. From Keycloak administrator console, in the upper right corner, click **Admin**:
 - b. Select **Manage account -> password**
3. Under **Clients -> Clara -> roles tab**, click the **Add role** button
4. Provide the role name **admin** and click **save**
5. Under **users**, click the **add user** button
6. Provide a user name and click **save**
7. Under **Credentials**, provide a password for the user, turn the **temporary** field to **off**, click the **Reset Password** button and confirm
8. Under **Role Mappings**, in the **Client Roles** dropdown, select **Clara**. Some boxes appear on the right

9. Under **Available Roles**, select **admin** and click the **Add Selected** button. The **admin** role appears in the **Assigned Roles** box
10. On the left navigation bar, select the **Realm Settings** page and go to the **Themes** tab
11. In the **Login Theme** parameter, select the Keycloak theme, then click **save**

Customizing SSL certificates

To install your own SSL certificates, run the following procedure:

1. In the `<install_path>/clara/nginx/cert` folder replace the `.key` and `.crt` default certificate files with your own files (do not change the default names).
2. Complete the installation procedure, or run the following commands from the `<install_path>` directory to update a pre-existing installation:

```
docker stop clara-nginx
```

```
docker start clara-nginx
```

Configuring Clara to use TLS

By default, Clara external port is configured to use TLS 1.2 and TLS 1.3.

To modify this setting, you can proceed in one of the following ways::

- Before launching Clara, build a tailored version of the `clara-nginx` image.
After extracting `clara-img.tar.gz` file (see installation instructions), go in the folder `clara-img/nginx` and edit the file `nginxAuth.conf`.
Configure the key `"ssl_protocols"` to the desired value (see nginx official documentation for more details).
- 1. On a running Clara, run the bash in the running container `clara-nginx`:

```
docker exec -ti clara-nginx bash
```


edit `/etc/nginx/conf.d/default.conf` file and configure the key `"ssl_protocols"` to the desired value (see nginx official documentation for more details).
Restart the container:

```
docker restart clara-nginx
```

How to set a custom client secret

1. From the Keycloak administration console, in the left side navigation bar, select **Clients->Clara**.
2. From the tab **Credentials**, click **Regenerate Secret**.
3. Copy the content of the field **Secret**.
4. In the `.env` file you find in `<BUILD_DIR>/clara/.env`, paste the content you copied from the field **Secret** into the `CLIENT_SECRET` parameter, overriding the default value.
5. From the `<BUILD_DIR>` directory, run the commands `./clara.sh --up --nc`.

Customizing Clara Knowledge Base

You can create a custom Knowledge Base using the Smart FAQ functionality. You can create Smart FAQs in two ways:

- By using the Smart FAQ user interface
- Importing the Smart FAQs from an Excel file

Using the Smart FAQ User Interface

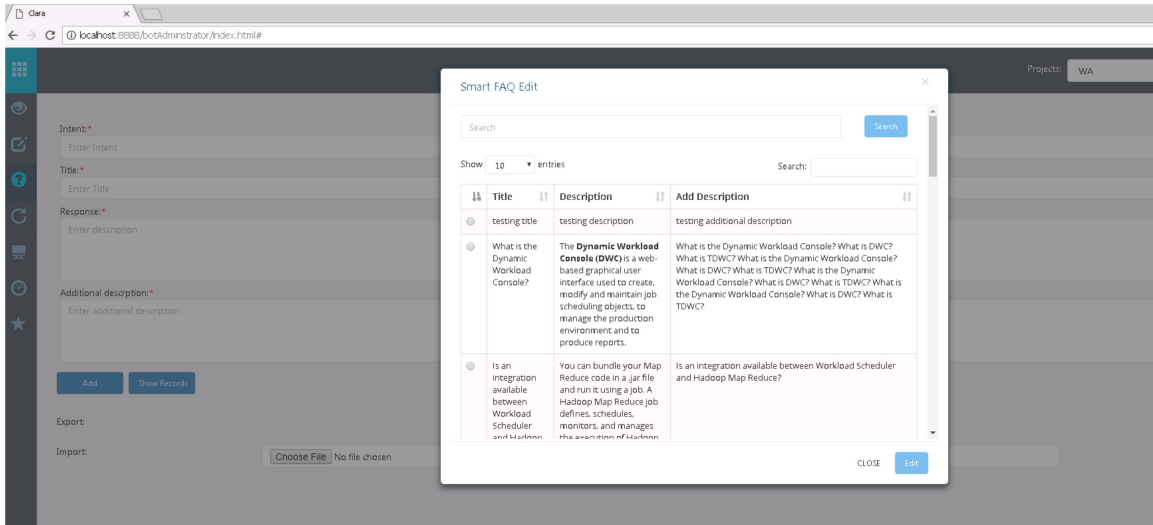
To create Smart FAQs by using the Smart FAQ user interface, run the following procedure:

1. Connect to Clara Administration Console https://<Clara_machine_IP>/botAdministrator/index.html
2. Open the Workload Automation project
3. Click the **Smart FAQ** tab and fill in the following text boxes:
 - Intent
 - Title
 - Description
 - Additional description
4. Click the **Add** button



5. You can see the added records by clicking the **Show Records** button

Installation Guide



6. You can test the Smart FAQs by entering **testing** in the chatbot. It will connect the FAQ bot and will provide an answer.



Importing the Smart FAQs from an Excel file

To import Smart FAQs from an Excel file, from the Clara Administration Console, run the following procedure:

1. Click the **Choose file** button and select the Excel file containing your Knowledge Base
2. To upload the file, click the **Import** button.

Installation Guide

The screenshot shows a web application interface for managing Smart FAQs. On the left is a dark sidebar with navigation icons. The main area contains a form with the following fields:

- Intent:** A text input field with the placeholder "Enter Intent".
- Title:** A text input field with the placeholder "Enter Title".
- Response:** A large text area with the placeholder "Enter description".
- Additional description:** A large text area with the placeholder "Enter additional description".

Below the form are two buttons: "Add" and "Show Records". At the bottom, there are "Export" and "Import" buttons. The "Import" button is next to a file input field that contains the text "Choose File" and "FAQ_WA_JBM.csv".

3. In the same way, you can export the Smart FAQs into an Excel file by clicking the **Export** button.

This screenshot is identical to the one above, showing the same Smart FAQs management interface. The "Export" button is now highlighted in blue, indicating it has been clicked. The "Import" button and file input field remain visible at the bottom.

Appendix

Installing Docker and Docker Compose

To remove any previous installation and reinstall Docker, run the following commands as a user with root privileges (i.e. sudo):

1. `systemctl stop docker`
2. `yum remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-selinux docker-engine-selinux docker-engine`
3. `rm /etc/yum.repos.d/docker*.repo`
4. `yum install -y yum-utils device-mapper-persistent-data lvm2`
5. `yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo`
6. `yum install docker-ce docker-ce-cli containerd.io`
7. `systemctl start docker`
8. `usermod -aG docker YOUR_DOCKER_USER_HERE`
9. `systemctl enable docker.service`
10. `systemctl daemon-reload`

For Docker Compose, run the following commands:

1. `curl -L https://github.com/docker/compose/releases/download/1.24.0/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose`
2. `chmod +x /usr/local/bin/docker-compose`
3. `sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose`

How to enable persistent notifications in Firefox and Chrome

Clara can monitor a job and send you a **notification** when the job status changes.

The first time you open Clara UI from a browser, you will be asked to allow Clara to send notifications. You can enable notifications in Google Chrome, Mozilla Firefox, or Microsoft Edge.

If you are enabling notifications in Firefox, to have persistent notifications outside the browser (like in Google Chrome), run the following procedure:

1. Open Firefox and type **about:config** in the address bar and press Enter. A warning message will appear, click the button "I accept the risk!".
2. Type "alerts" in the Search filter box: the preference **alerts.useSystemBackend** will show up in the window.
3. Set the preference to true if you want to have websites notifications in Windows 10 Action Center.

If you want the notification to be persistent inside Firefox notification system, run the following procedure:

1. Open Firefox, type **about:config** in the address bar, and press Enter. A warning message will appear, click the button "I accept the risk!".
2. In the Search filter box, type **dom.webnotifications.requireinteraction.enabled**. The preference will show up in the window.
3. Set the preference to true.

If you want to receive notification in Google Chrome on Mac:

1. Got to System Prefecences > Notifications
2. Select Google Chrome in the list of browsers
3. Turn on the notification toggle

User's Guide

About the User's Guide

The User's Guide provides information about how to use HCL Clara.

[Managing User Credentials](#)

[Interacting with Clara](#)

[Clara replies to your questions](#)

[Clara runs actions on your behalf](#)

[Clara monitors your servers through HERO](#)

[Clara helps in problem determination](#)

For detailed user scenarios, see the **videos**:

[Clara for Workload Automation](#)

[Clara for Workload Automation for Z](#)

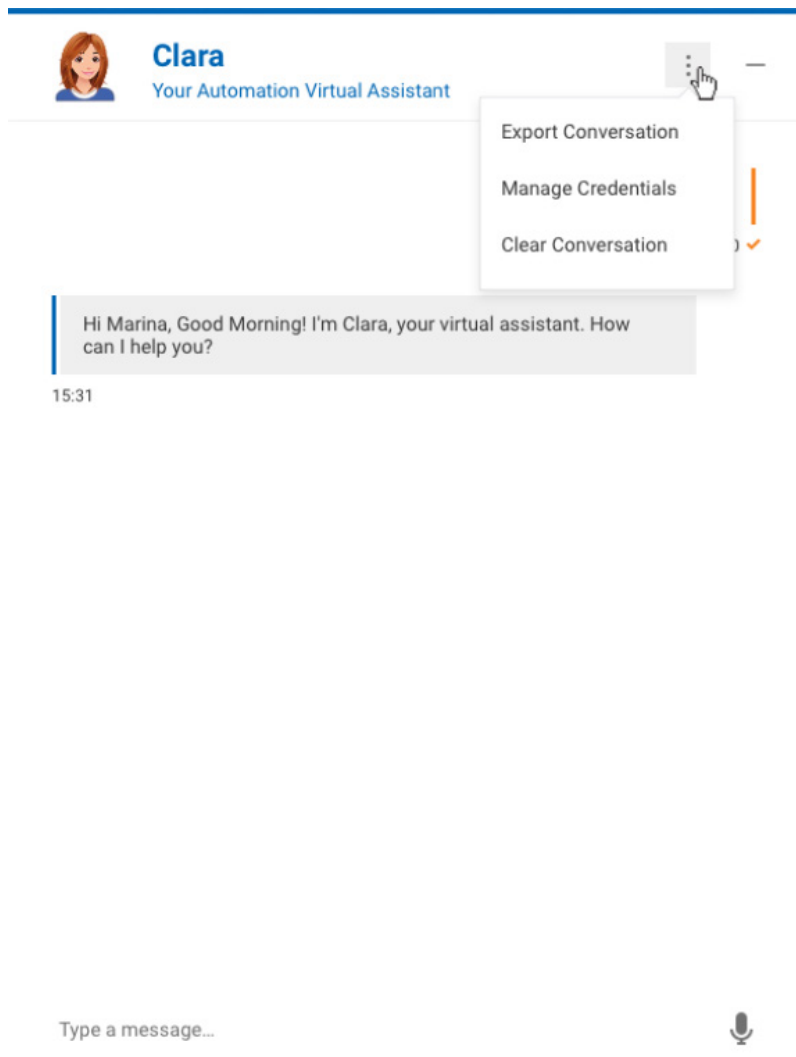
Managing User Credentials

The User Credential Manager component helps you map your Clara credentials with Workload Automation credentials. Each Clara user can only manage his own Workload Automation credentials.

As you start chatting with Clara, you have the option to manage credentials for your product environments. Valid credentials are needed to invoke the REST APIs of the target product environment, thus enabling Clara to run actions on your behalf.

To manage user credentials, run the following steps:

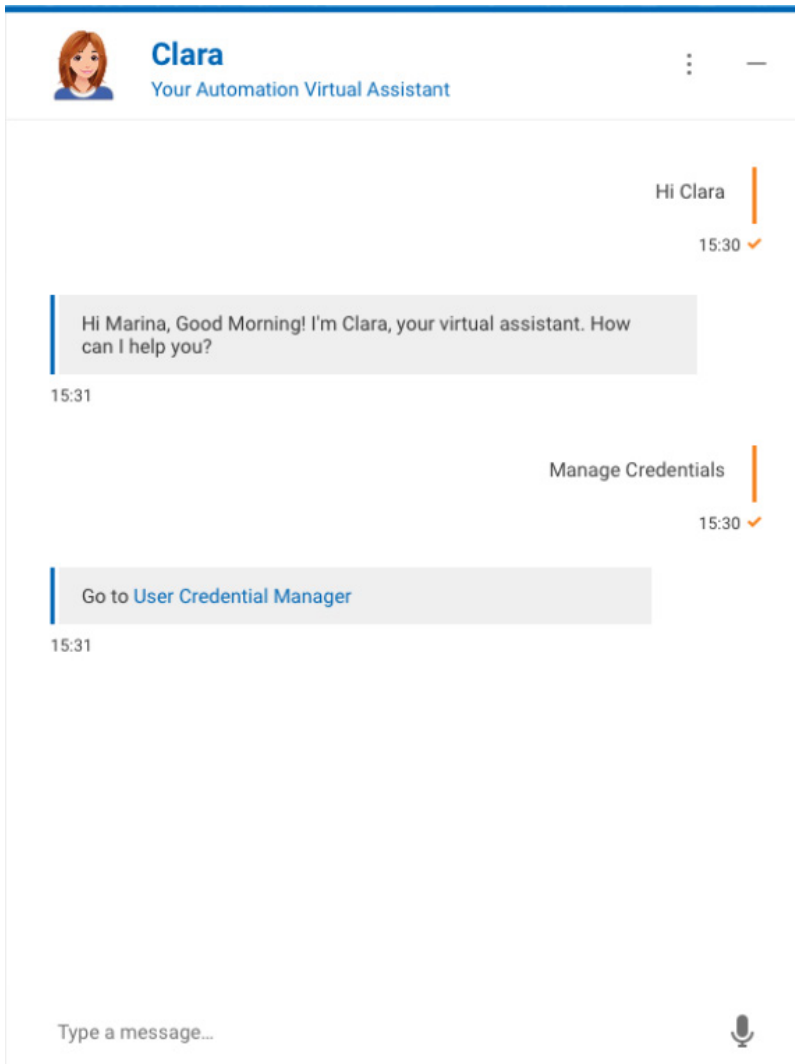
1. Open Clara Web Interface.
2. Click the menu icon in the upper right corner and select **Manage Credentials**.



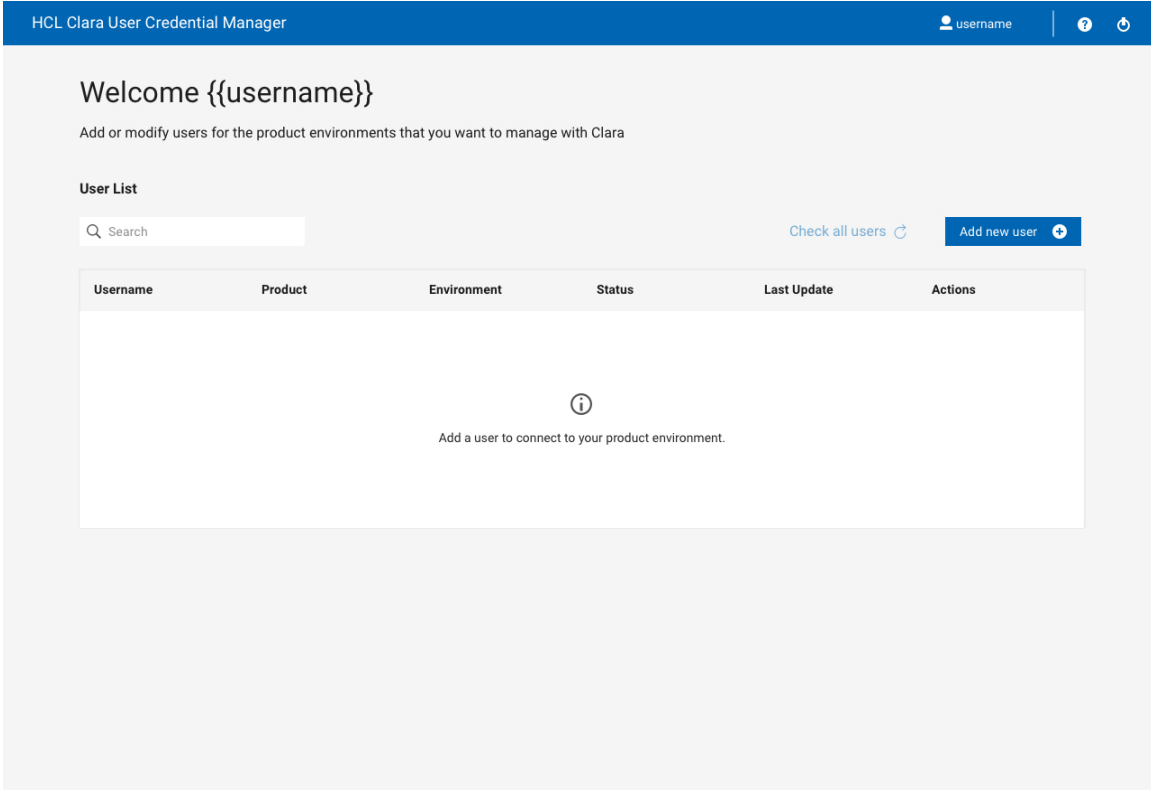
As an alternative, type one of the following messages:

- credential manager

- add user credentials
 - change creds
3. Clara provides the link to the **User Credential Manager**.
 4. Moreover, when you ask Clara to run actions on your target environment but your credentials are not set, or not valid, Clara alerts you and provides the link to open the User Credential Manager interface.
 5. In any case, click the link:

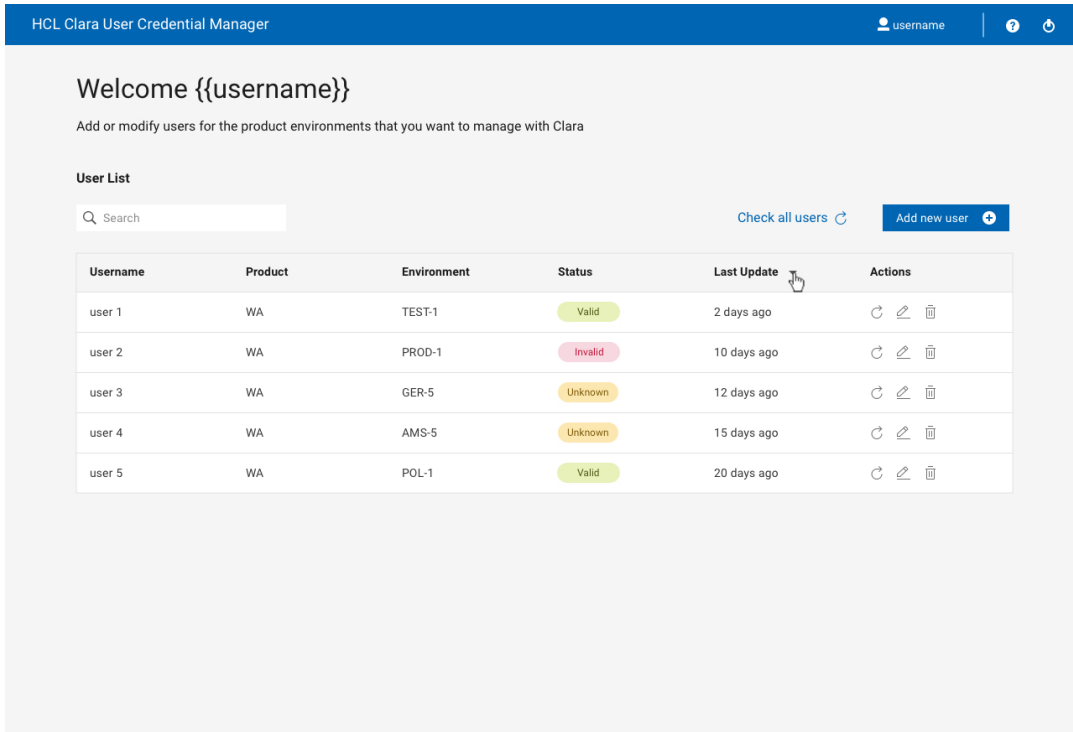


6. The User Credential Manager interface opens:



Click **Add new user** to add a valid user for each of your different product environments. For details, see [Adding a user](#).

Once you have added users for all your environments, the User Credential Manager main page will look like the following:



If the status of some user credentials is **Unknown**, click **Check all users** to validate user credentials and update their status.

7. In the User Credential Manager main page, you can find the following information:
 - The users you have defined to enable Clara managing your product environments.
 - For each product environment, the status of your user credentials. It can be one of the following:
 - **Unknown** - Clara has not tried to log in to your product environment yet, or she tried but was not able to connect
 - **Valid** - Clara logged in to your product environment successfully
 - **Invalid** - Clara was not able to log in to you product environment
 - The number of days since the last update of the user credentials for your product environment.
8. From the User Credential Manager main page, you can run the following actions:
 - Add or modify users for each product environment.
 - Check if user credentials are valid.
 - Delete user credentials.

Adding a User

You can add a user for the product environment that you want to manage with Clara. You can only have one user for each product environment.

To add a new user, run the following steps:

1. From the User Credential Manager main page, click the button **Add new user**
2. The Add User panel opens:

HCL Clara User Credential Manager username | ? ⏻

Add User

Add a user for the product environment that you want to manage with Clara
You can only have one user for each product environment.

Account Management

Product: WA Environment: PROD-1

Username: _____ Password: _____

Test

Personal Information

Email: email@address.com

Cancel Save

3. Select Product and Environment.
4. Provide Username and Password for the environment.
5. Click Test to check user credentials.
6. Provide your e-mail address to receive notification e-mails from Clara.
7. If user credentials are **Valid**, click Save.
8. If it is not possible to check user credentials because, for example, the connection with the target environment is down, you can save the new user with **Unknown** status for credentials and verify them later on.
9. You can save the user even if credentials are **Invalid**, and modify them later on.

Checking credentials

Besides checking all user credentials at one time by clicking the **Check all users** button, you can check credentials for a single user. Run the following steps:

1. From the User Credential Manager main page, select the **check credentials** action for the user whose credentials you want to verify:

HCL Clara User Credential Manager username | ? ↻

Welcome {{username}}

Add or modify users for the product environments that you want to manage with Clara

User List

🔍 Search Check all users ↻ [Add new user](#) +

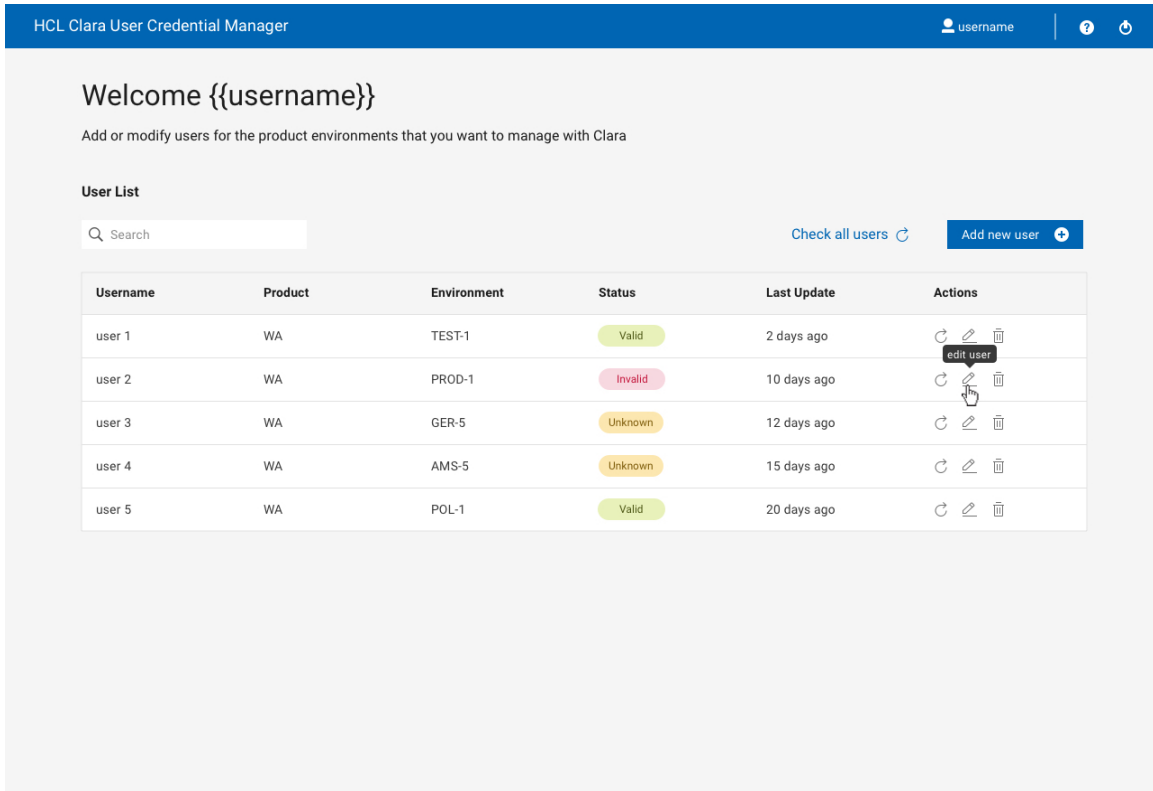
Username	Product	Environment	Status	Last Update	Actions
user 1	WA	TEST-1	Valid	2 days ago	↻ ✎ 🗑️
user 2	WA	PROD-1	Invalid	10 days ago	↻ ✎ 🗑️
user 3	WA	GER-5	Unknown	12 days ago	↻ ✎ 🗑️
user 4	WA	AMS-5	Unknown	15 days ago	↻ ✎ 🗑️
user 5	WA	POL-1	Valid	20 days ago	↻ ✎ 🗑️

2. Verify if the status of your user credentials changes.
3. If you need to modify your user credentials, select the **edit user** action.

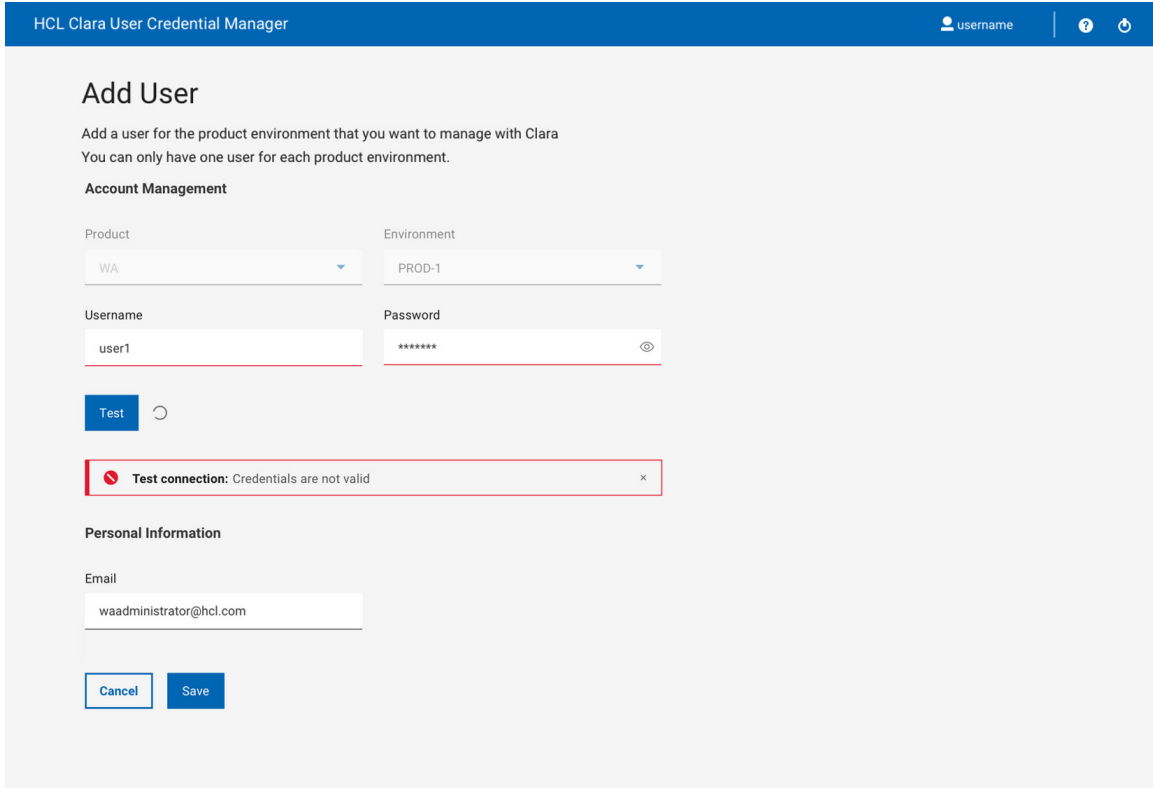
Editing a user

To edit a user, run the following steps:

1. From the User Credential Manager main page, select the **edit user** action for the user you want to modify:



2. The Edit User panel opens:



3. Modify Username, Password, or e-mail information.

4. Click Test to check the user credentials.
5. If the user credentials are **Valid**, click Save.
6. You can save the user even if credentials are **Invalid** or their status is **Unknown**, and modify them or check their status later on.

Deleting a user

To delete a user, run the following steps:

1. From the User Credential Manager main page, click the **delete user** action for the user you want to remove.

The screenshot shows the HCL Clara User Credential Manager interface. At the top, there is a blue header with the text "HCL Clara User Credential Manager" on the left and a user profile icon with the text "username" on the right. Below the header, there is a "Welcome {{username}}" message and a sub-header "Add or modify users for the product environments that you want to manage with Clara". The main content area is titled "User List" and contains a search bar, a "Check all users" link, and an "Add new user" button. Below this is a table with the following columns: Username, Product, Environment, Status, Last Update, and Actions. The table contains five rows of user data. A tooltip labeled "delete user" is visible over the delete icon in the Actions column for the second row (user 2).

Username	Product	Environment	Status	Last Update	Actions
user 1	WA	TEST-1	Valid	2 days ago	
user 2	WA	PROD-1	Invalid	10 days ago	delete user
user 3	WA	GER-5	Unknown	12 days ago	
user 4	WA	AMS-5	Unknown	15 days ago	
user 5	WA	POL-1	Valid	20 days ago	

2. In the confirmation panel, click the Delete button to permanently remove the user from the system.

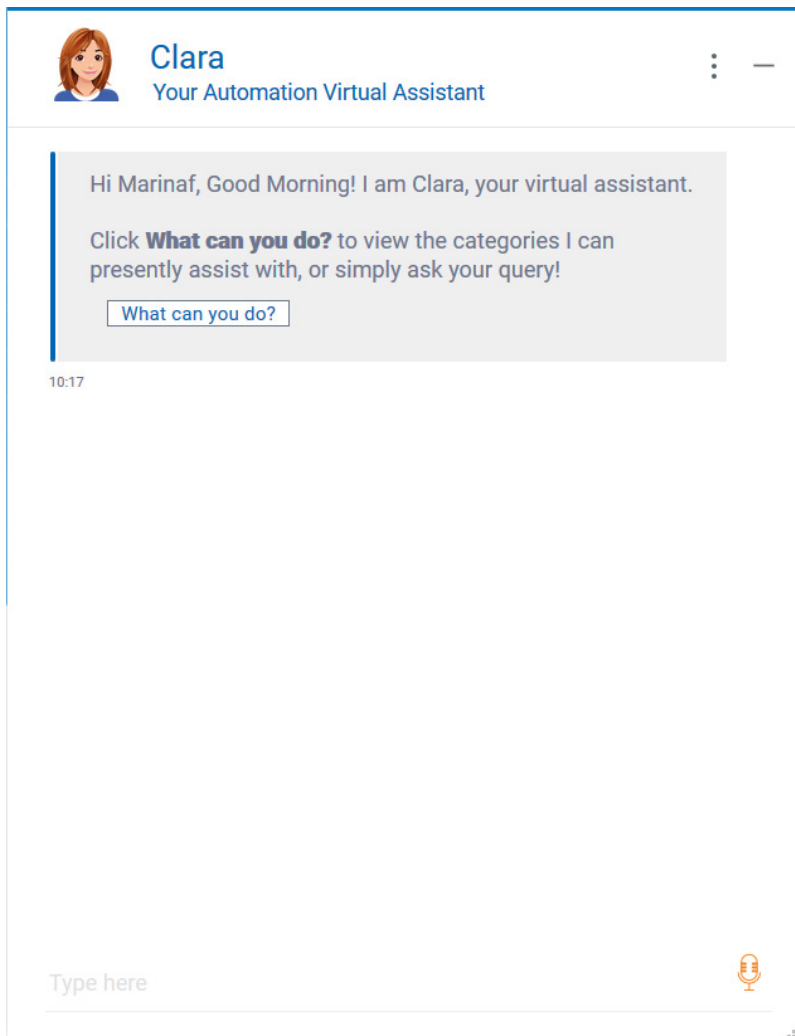
The screenshot shows the HCL Clara User Credential Manager interface. At the top, there is a dark blue header with the text 'HCL Clara User Credential Manager' on the left and a user profile icon labeled 'username' on the right. Below the header, a grey area contains a welcome message: 'Welcome {{username}}' followed by the instruction 'Create and manage all your access credentials enabling Clara to perform actions'. A section titled 'User List' features a search bar, a 'Check all users' button, and an 'Add new user' button. A table with columns 'Username', 'Product', 'Last update', and 'Actions' is displayed. A modal dialog box titled 'Confirm Delete' is overlaid on the table, containing the text: 'All data will be permanently removed from the system. Are you sure you want to delete the user?'. The dialog has 'Cancel' and 'Delete' buttons at the bottom.

Username	Product	Last update	Actions
user 1	WA	ago	
user 2	WA	ago	
user 3	WA	ago	
user 4	WA	ago	
user 5	WA	ago	

Interacting with Clara

You can contact Clara by using one of the following options:

- Clara web interface, available at the following link <https://<IP:PORT>/Clara/>
- Clara embedded in the Dynamic Workload Console of Workload Automation
- Messaging tools such as Slack and Sametime.

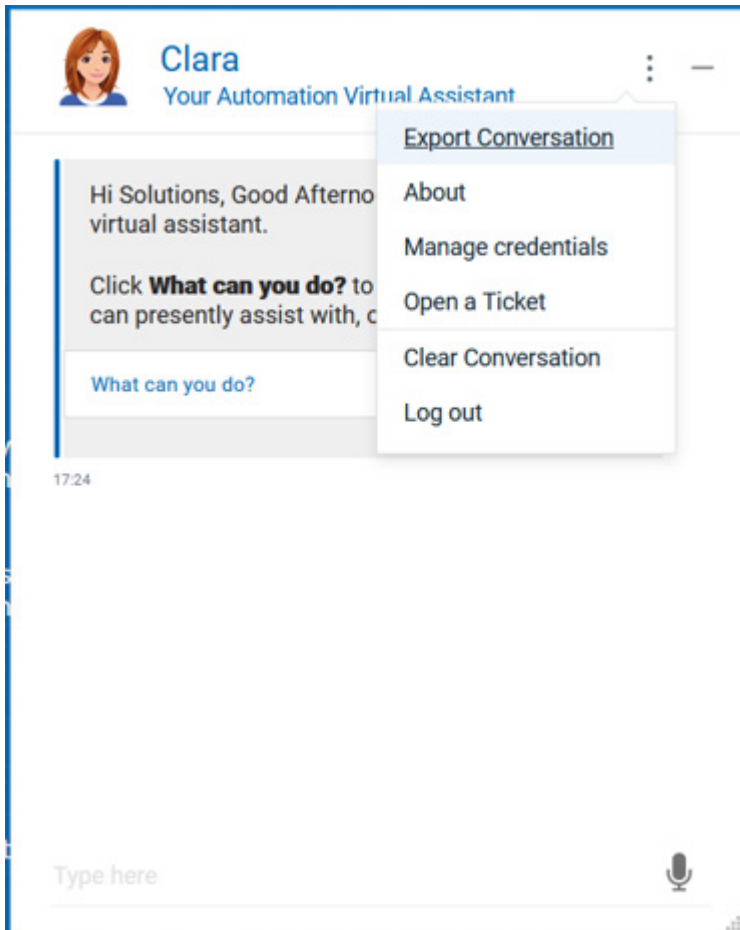


Clara web interface at startup

At Clara startup, you can:

- Ask Clara "*What can you do?*" to view the categories Clara can assist with. For an overview of the categories Clara can assist with, see [Running actions on your behalf](#).
- Simply ask your free question (for example, *what is a job? what is a job stream?*). For details, see [Clara replies to your questions](#).
- Ask Clara to help you in problem determination (for example, *find out why a job does not start*). For details, see [Clara helps in problem determination](#).

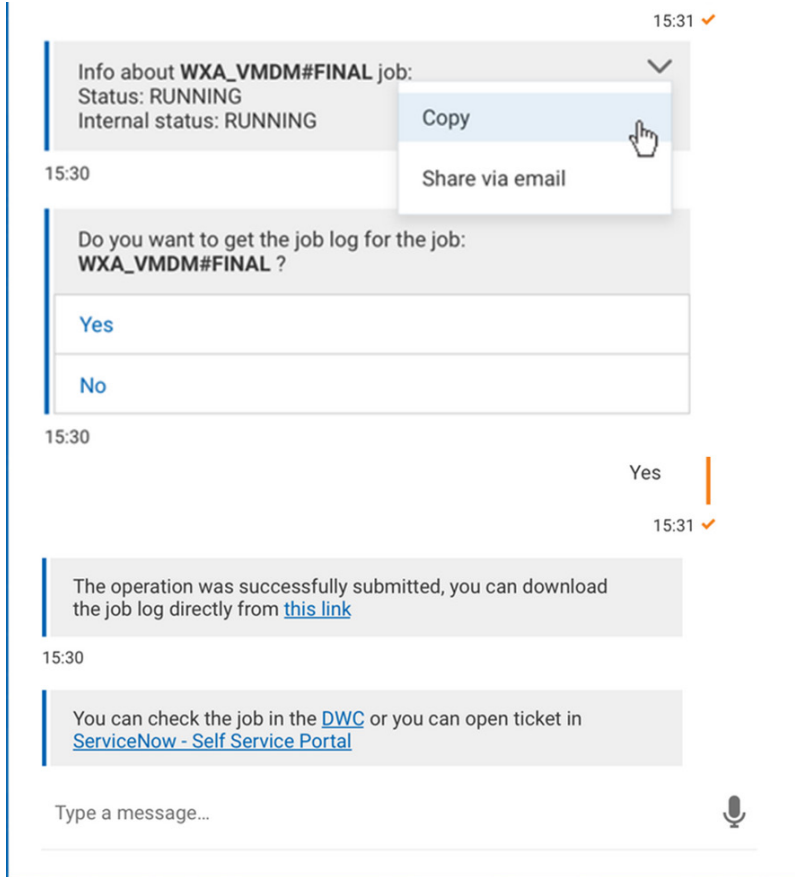
By clicking the menu icon on the upper right corner of Clara chat window, you can find the following options:



- Export Conversation - to export the current conversation with Clara
- About - to get information about Clara version, release, and terms of license agreement.
- Manage credentials - to map your Clara credentials with Workload Automation credentials
- Open a Ticket - to get the link to the Service Platform of your choice
- Clear Conversation - to reset the current conversation with Clara
- Log out - to log out from Clara

Copy and share Clara's messages via email

When chatting with Clara, by hovering over Clara's messages, options are available to **Copy** or **Share via email** the content of the message.



By selecting **Copy**, you can copy the message to clipboard, and paste it later on.

By selecting **Share via email**, the message text is copied into a new e-mail in your default e-mail program.

Clara replies to your questions

Clara is the product expert that you can consult to learn the product capabilities.

If your question is not clear enough to Clara, she displays several choices and asks you to select one. You can also select: "None of above", and Clara will provide additional options.

Semantic search

If you select "None of above" twice in a row, to increase the response rate Clara uses an advanced **semantic search** engine combined with machine learning technologies to search in two different sources of information:

- Online [Workload Automation documentation](#)
- [Automation Hub](#)

and provides links to the 5 most relevant search results.

You can even ask for a semantic search directly, by typing "Search" followed by the subject you want to search.

Example: *Search how to be notified if a job fails.*

Clara can also help you understand the meaning of Workload Automation error messages: ask Clara "Search <error_message_code>", or "Search error <error_message_code>" to get an explanation of the error message you've received.

Example of FAQs from a Workload Automation Operator

- *What is a job?*
- *What is a job stream?*
- *What are the benefits of WA?*
- *How can I create a job?*
- *How can I export a WAT?*
- *What is a dynamic agent?*
- *What is the different between FTA and dynamic agent?*

Example of FAQs from a Workload Automation Administrator

- *What is the latest WA version?*
- *What are the main feature of the latest WA version?*
- *Does WA support docker?*
- *Does WA support the integration with Microsoft Azure?*
- *What are the default roles available in the DWC?*

- *How can I send an email if a job fails?*
- *What is the command to switch from MDM to BKM?*

Clara
Your Automation Virtual Assistant

what is a job stream? 19:49 ✓

A **Job Stream** is a sequence of jobs to be run, together with times, priorities, and other dependencies that determine the order of processing. Each job stream is assigned a time to run, represented by run cycle with type calendar, set of dates, or repetition rates.

19:49

Do you want to know how to create a Job Stream? 19:49

yes 19:49 ✓

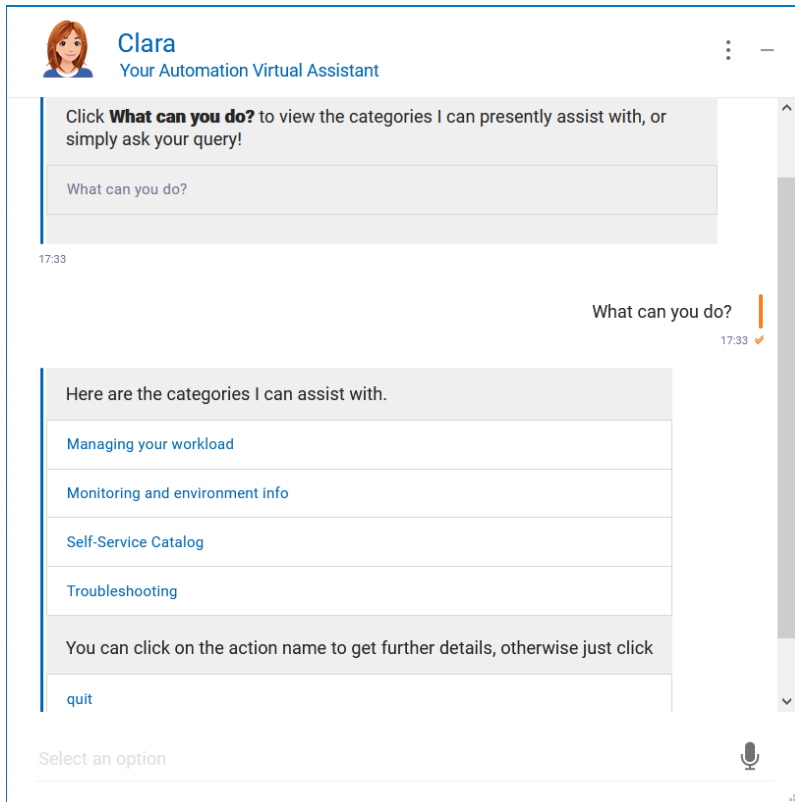
To create a Job Stream from the web console, search for the page 'Manage Workload Definition'.

Search bar: Default Administration Planni
manage w
Use ALT-F3 on your keyboard to open this search box.
Manage Workload Reports

[type here]

Clara runs actions on your behalf

At Clara startup, by entering the "What can you do?" conversation flow, you can view the actions that Clara can run on your behalf.



The actions that Clara can run on your behalf, for WA and WAZ, are grouped in the following categories:

1. **Managing your workload** - You (as an operator) can submit a job or modify a job execution
 - **Rerun a job** - rerun a job that is in error status (available for WA only)
 - **Submit a job** - submit a predefined job (available for WA only)
 - **Submit a jobstream** - submit a predefined jobstream for processing (available for WA and WAZ)
 - **Cancel a job** - cancel a job before it is launched (available for WA only)
 - **Release a job** - release a job from dependencies (available for WA and WAZ)
 - **Put a job in HELD status** - put a job in HELD status, awaiting for dependencies resolution (available for WA and WAZ)

2. **Monitoring and environment info** - You (as an administrator) can get information about scheduled jobs
 - **Monitoring the scheduling environment** - to get information about your jobs and the daily plan
 - **Get the status of a job** - retrieve information about the current status of a job (available for WA and WAZ). For details, see [Get the status of a job](#).
 - **Monitor a job** - monitor a job and receive a notification when the job status changes (available for WA only). For details, see [Monitor a job](#).
 - **Get the overall status of the daily plan** - retrieve information about the actual status of all the jobs in the daily plan (available for WA and WAZ). For details, see [Get the status of the daily plan](#).
 - **Monitoring WA servers through HERO** - to monitor and get information about your WA servers through HERO (available for WA only). For details, see [Clara monitors servers through HERO](#).
 - **Get the WA version/release in your environment** - retrieve information about the current version or release of Workload Automation installed in your environment (for WAZ, the information retrieved are related to the Dynamic Workload Console)
3. **Self-Service Catalog** - You (as an operator or administrator) can manage workload from Self-Service Catalog
 - **Submit a service from the catalog** - (available for WA and WAZ). For details, see [Submit a service from Self-service Catalog](#).
4. **Troubleshooting** - You (as an operator or administrator) can speed up troubleshooting WA issues. For details, see [Clara helps in problem determination](#).
 - **Why a job doesn't start?** - You can understand why a job is not starting (available for WA only)
 - **Are there any jobs in error?** - You can find any jobs in error in the daily plan (available for WA and WAZ)
 - **Are there any jobs in late?** - You can find any jobs in late in the daily plan (available for WA and WAZ)
 - **Open a ticket** - Clara provides you with a link to the Service Platform of your choice (available for WA and WAZ). For details, see [Open a ticket](#).

When you run an action, Clara asks you in which Workload Automation environment you want to run it. If you have only one environment configured, Clara will choose that one.

For each action in the **Managing your workload** category, Clara asks for a justification. This helps store information in the audit store of Workload Automation where the administrator can check who is the operator that performed a specific action.

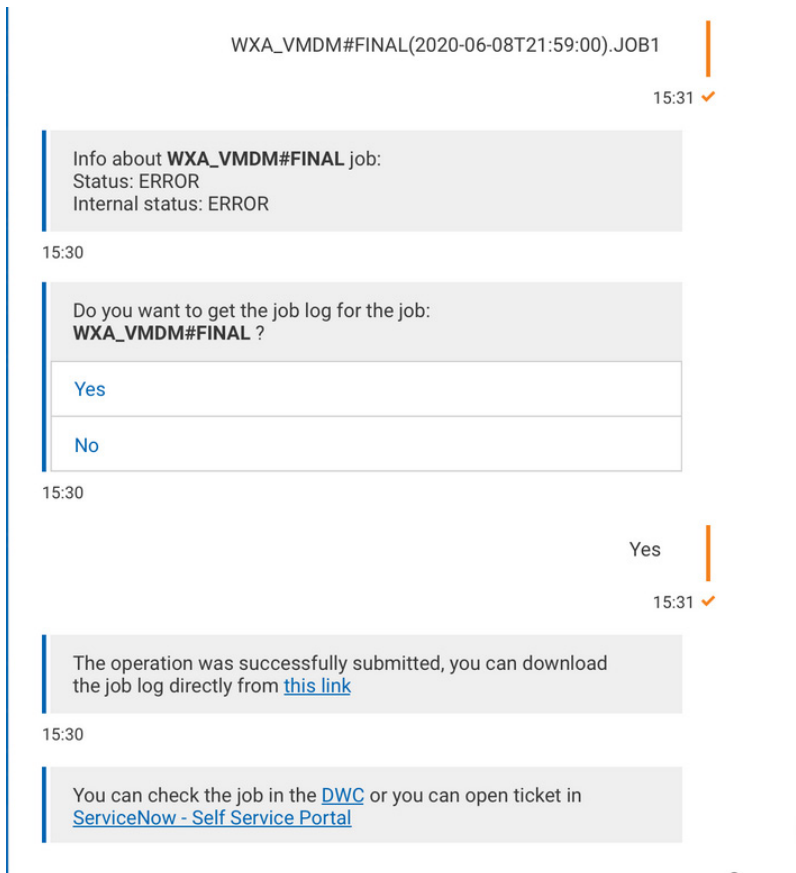
Also, Clara leverages the authentication mechanism of Workload Automation. If the user is not authorized to run an action, Clara returns an error. You can also extend to Clara the authorization to run actions in your place. For details, see [Managing User Credentials](#).

Get the status of a job

By selecting this action you ask Clara to provide the current status of a job.

After providing the job status, for job statuses different from SUCCESSFULLY COMPLETED, Clara provides you with a link to download the job log. Also, Clara provides you with a link to the Service Portal of your choice in case you need to open a Service Ticket. For details, see [Open a ticket](#).

For all job statuses including SUCCESSFULLY COMPLETED, Clara provides you with a direct link to the Dynamic Workload Console - Monitoring Workload - where a pre-set query will show you details about the job. This Clara functionality prerequisites the DWC V9.4 or the DWC V9.5 (only FP3 or higher).

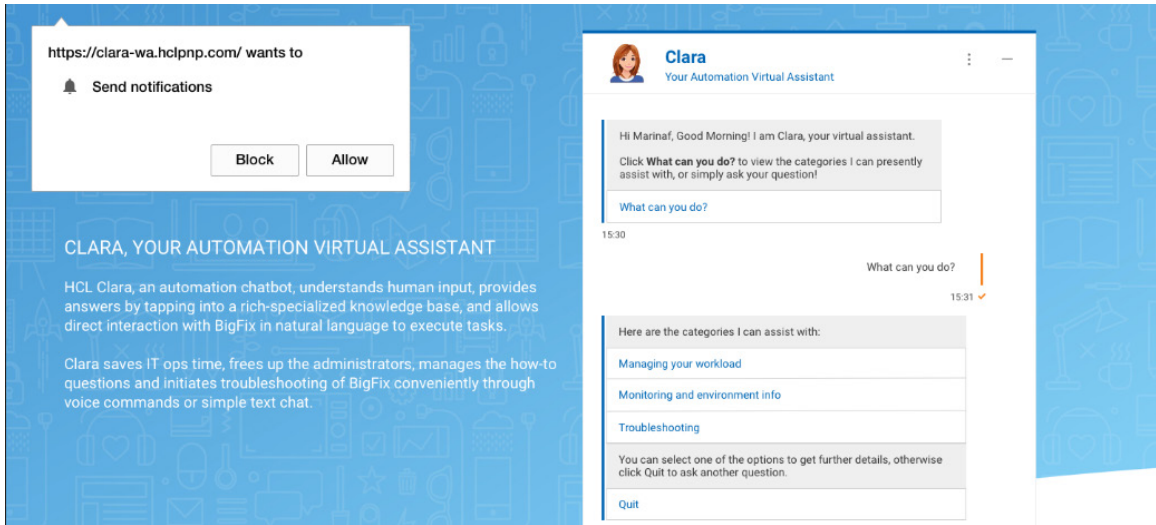


Monitor a job

Clara can monitor a job and send you a **notification** when the job status changes.

The first time you open Clara UI from a browser, you will be asked to allow Clara to send notifications. See instructions about [How to enable persistent notifications](#).

Clara checks the job status at time intervals, depending on the estimated duration of the job in Workload Automation, and notifies you when the job status changes. An e-mail with the new job status is also sent to the e-mail address you specified in the Add User flow, in the [User Credential Manager](#) interface.



If you don't allow notifications when opening Clara UI, Clara will assist you to enable notifications in you browser later on.

To monitor a job and receive Clara notifications, run the following steps:

1. In the "What can you do?" conversation flow, select **Monitoring and environment info > Monitor a job**
2. If not already done before, enable notifications in your browser by following the instructions provided by Clara.

The screenshot shows the Clara chat interface. At the top, there is a header with a profile picture of Clara, her name "Clara", and the subtitle "Your Automation Virtual Assistant". To the right of the header are three vertical dots and a minus sign. The main chat area contains three messages:

- Message 1:** A grey message box with the text "With the following options, I can help you get information about scheduled jobs." Below this are four blue links: "Get the status of a job", "Monitor a job", "Get the overall status of the daily plan", and "Get the WA version/release in your environment". A second grey message box follows, stating "You can select one of the actions to get further details, otherwise select one of the following options:". Below this are two blue links: "Back" and "Quit".
- Message 2:** A user message "Monitor a job" with a timestamp of "15:30" on the left and a confirmation mark "15:31 ✓" on the right.
- Message 3:** A grey message box with the text "I can monitor a job for you, and send you a notification when the status of the job changes." Below this is an **Important:** notice: "Make sure you enabled notifications for this site." Two blue links follow: "How can I enable notifications?" and "Continue with Monitor a job".
- Message 4:** A user message "How can I enable notifications?" with a timestamp of "15:30" on the left and a confirmation mark "15:31 ✓" on the right.
- Message 5:** A grey message box with the text "To enable notifications in your browser, see the following instructions:" followed by three blue links: "Enabling notifications in Chrome", "Enabling notifications in Firefox", and "Enabling notifications in Edge".

At the bottom of the chat area, there is a text input field with the placeholder "Type a message..." and a microphone icon to its right.

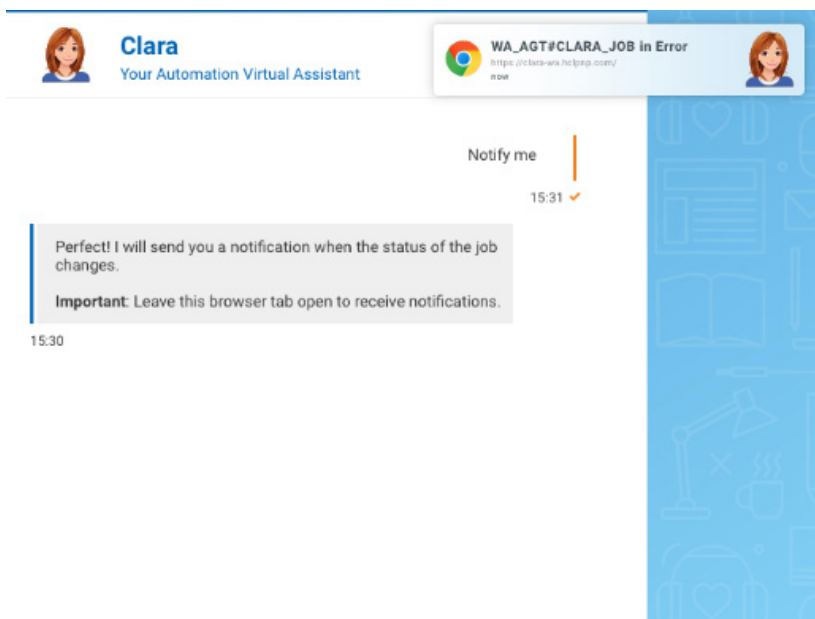
3. When required by Clara, provide the job name, the job stream name, and the workstation name for the job you want to monitor.
4. Clara will return the job status.

The screenshot shows a chat window for Clara, 'Your Automation Virtual Assistant'. The interface includes a header with Clara's profile picture and name, and a main chat area with a message history and a text input field at the bottom. The chat history consists of several messages:

- A user message: "Tell me the Job Stream Name" (15:30).
Clara's response: "WXA_VMDM#FINAL(2020-06-08T21:59:00).JOB1" (15:31) with a checkmark.
- A user message: "Tell me the Workstation Name" (15:30).
Clara's response: "WXA_VMDM#FINAL(2020-06-08T21:59:00).JOB1" (15:31) with a checkmark.
- A user message: "Info about WXA_VMDM#FINAL job:" (15:30).
Clara's response: "Status: RUNNING\nInternal status: RUNNING" (15:30).
- A user message: "Do you want to get the job log for the job: WXA_VMDM#FINAL ?" (15:30).
Clara's response: A list with "Yes" and "No" options (15:30).
- A user message: "Yes" (15:30).
Clara's response: "The operation was successfully submitted, you can check the job log directly in the [DWC](#)." (15:31) with a checkmark.
- A user message: "I can monitor this job for you and send you a notification when the status of the job changes." (15:30).
Clara's response: A list with "Notify me" option (15:30).

The bottom of the chat window shows a text input field with the placeholder "Type a message..." and a microphone icon.

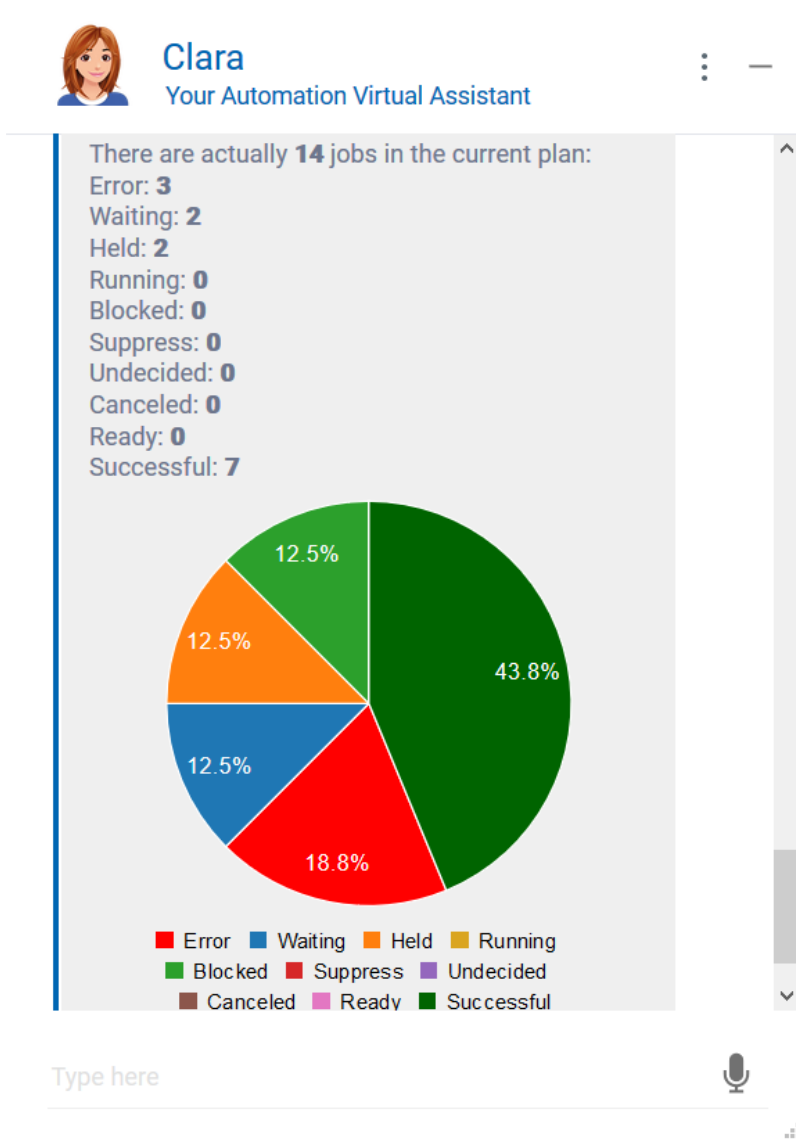
5. Leave the browser tab open. When the job status changes, Clara will send you a notification.



Get the status of the daily plan

As a Workload Automation administrator, you might need to check the status of the daily plan. Run the following steps:

1. In the "What can you do?" conversation flow, select **Monitoring and environment info > Get the overall status of the daily plan**
2. Clara provides an intuitive **graphical representation** of the status of jobs in the daily plan.

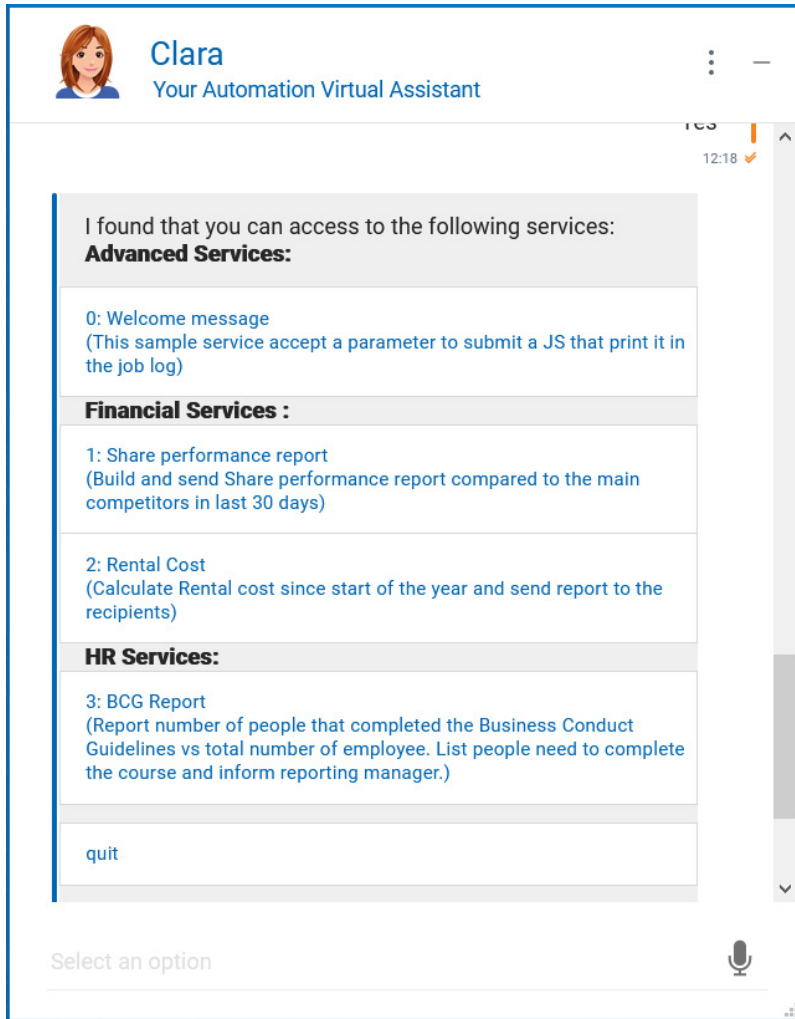


Submit a service from Self-service Catalog

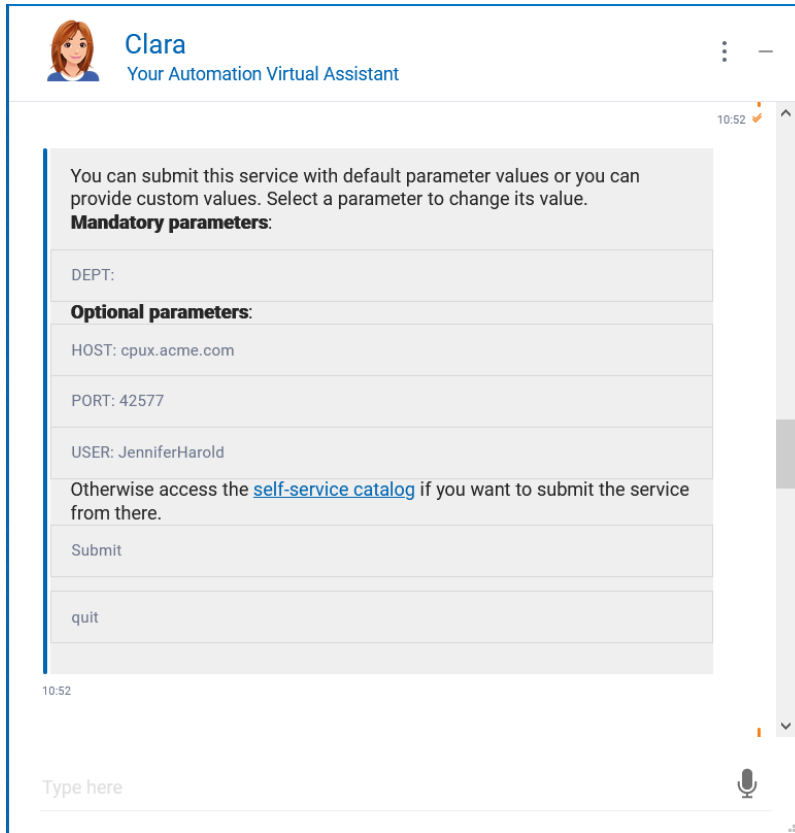
To select this action, ensure you have access to Self-Service Catalog, otherwise contact WA administrator.

When you select this action, Clara shows the services available to you in the Self-service Catalog.

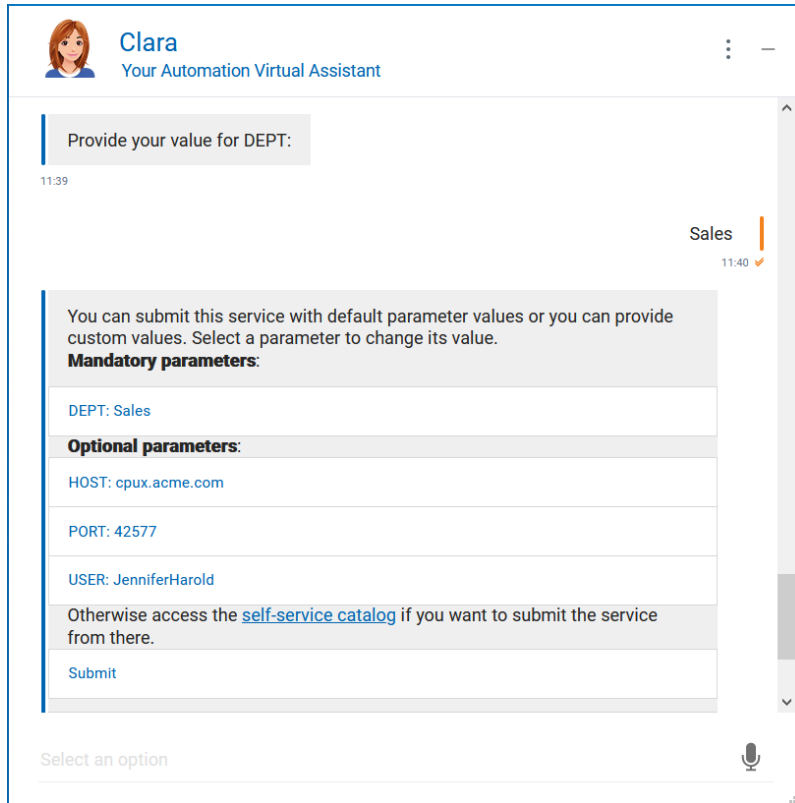
1. Select the service that you want to submit.



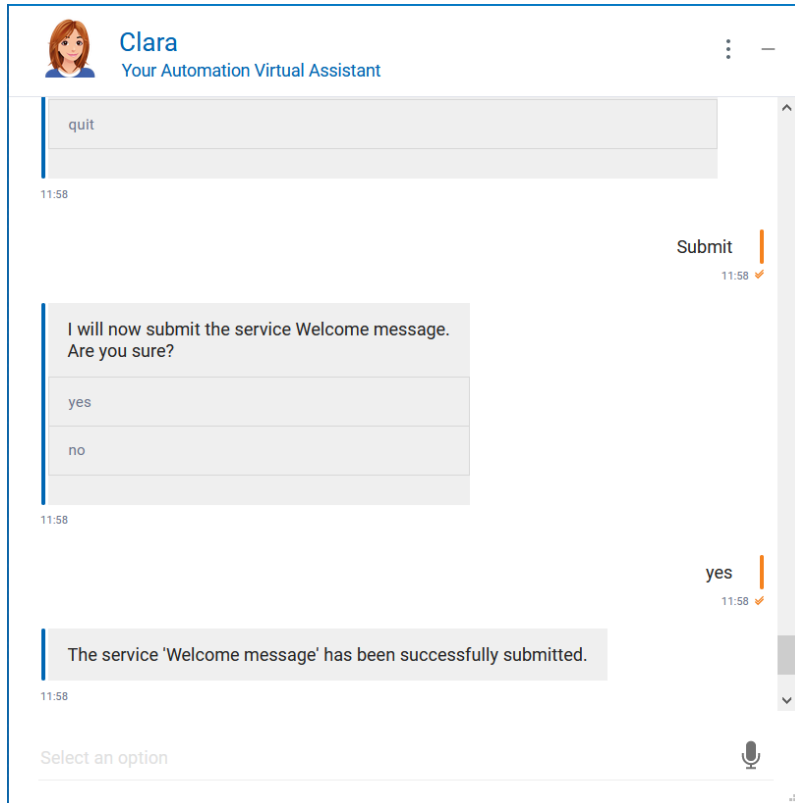
2. Mandatory and optional parameters are displayed by Clara with default values, if any.



3. You can submit the service with default parameter values or you can provide custom values. Select a parameter to set or change its value.

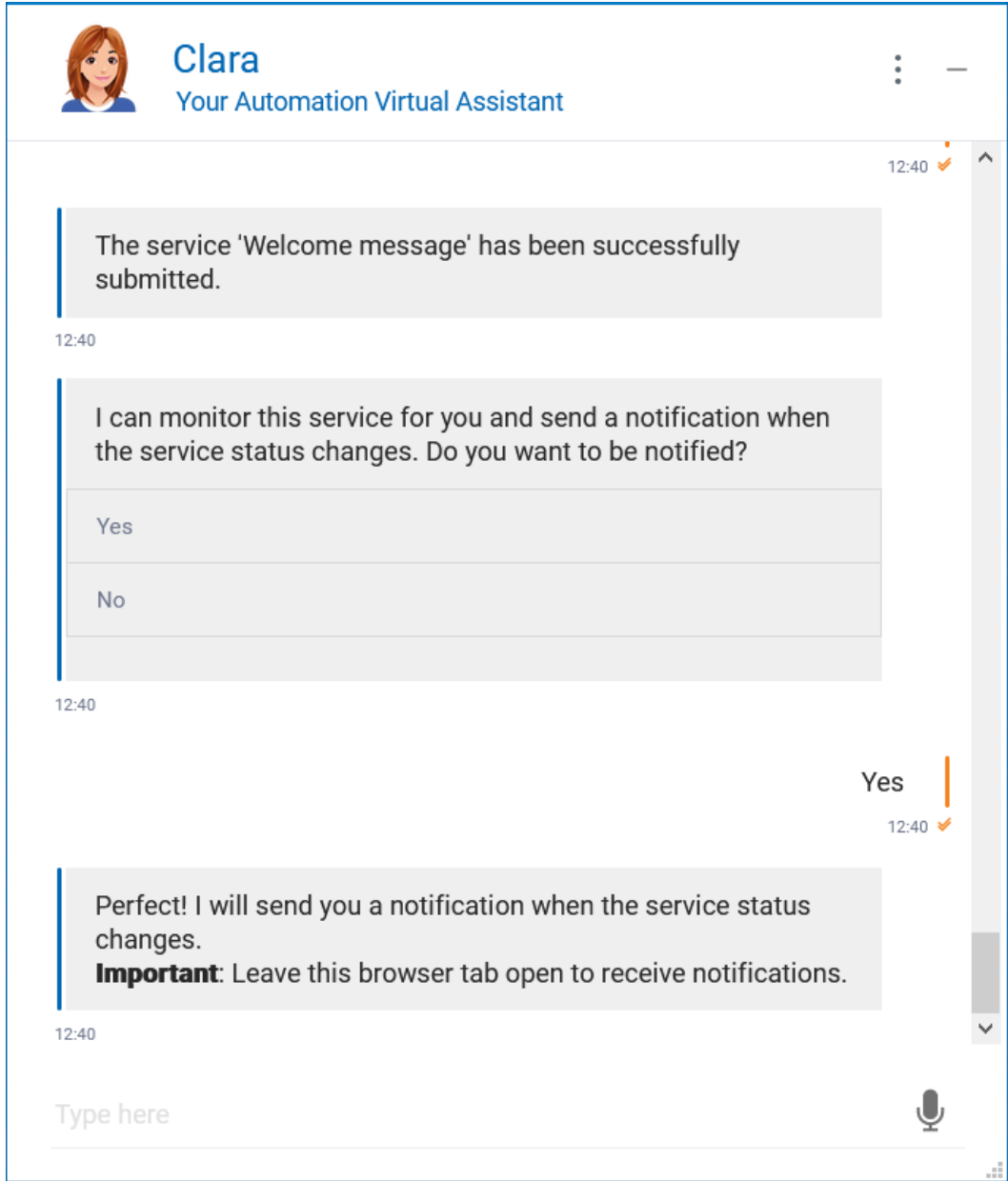


4. Click **Submit** to submit the service. When Clara asks for confirmation, click **Yes**. Also, Clara provides a link to the Self-service Catalog if you want to modify the parameters and submit the service from there.



5. Clara can monitor the service execution for you and, optionally, can send you a **notification** when the service status changes. Click **Yes** to receive notifications.

Note: Remember to allow Clara to send notifications. If you don't allow notifications when opening Clara UI, Clara will assist you to enable notifications in you browser later on.



6. Clara will notify you when the status of your submitted service changes into:

- Started
- Completed successfully
- Completed with errors
- Error
- In late

Clara monitors your servers through HERO

You can monitor and get information about your WA servers through **HERO (Healthcheck & Runbook Optimizer)**. This feature is available for WA users only (not WAZ).

HERO effectively helps **Workload Automation** Administrators monitor the health of their servers and perform informed recovery actions with specialized runbooks, keeping Workload Automation environments responsive and reliable. For additional information about HERO, see [HERO Product Overview](#).

From the *What can you do?* conversation flow, select **Monitoring and environment info > Monitoring WA servers through HERO**.

The following actions are available. Click on any option to see additional details. :

- [Get the status of your production servers](#)
- [Monitoring a specific server](#)
- [Are there any issues with my servers?](#)

With the following options I can help you monitor and get information about your WA servers through HERO.

- Get the status of all the PROD servers
- Monitoring a specific server
- Are there any issues with my servers?

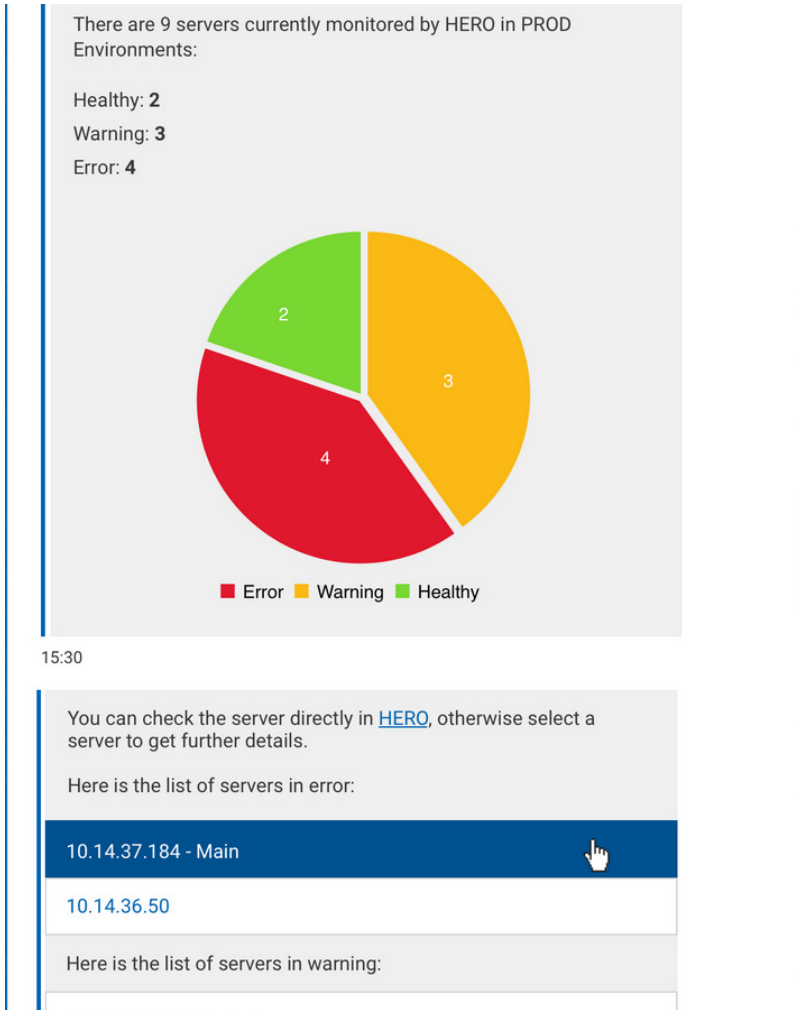
You can click on the action name to get further details, otherwise just click

Back

Quit

Get the status of your production servers

By selecting this action, you can retrieve information about the current status of Workload Automation **production servers**.



Clara shows a graph with a summary of the current status of your production servers. Status categories are: healthy, warning, error.

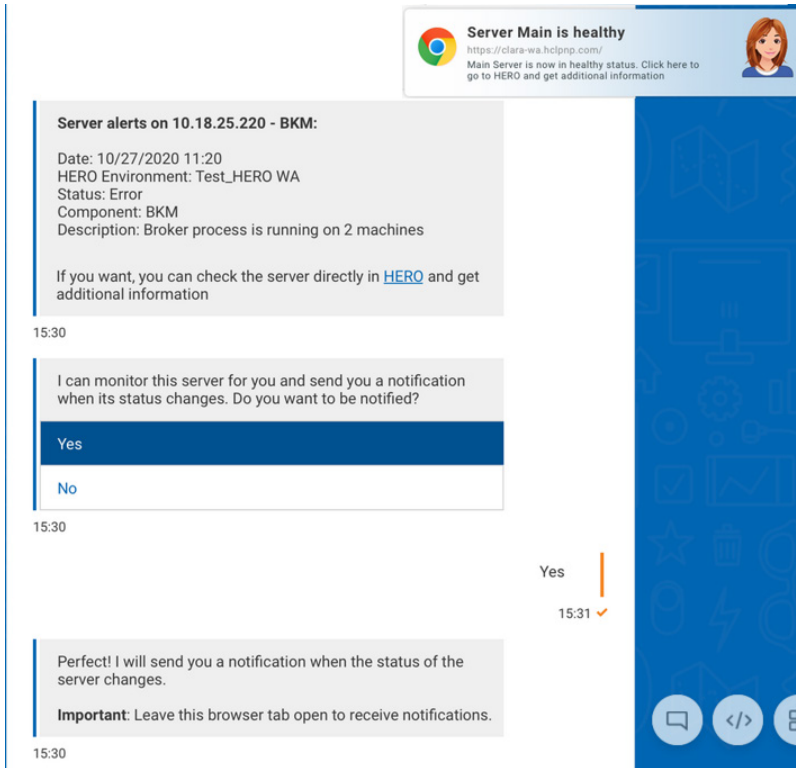
A direct link to HERO is available if you want to check any of your servers in HERO.

For servers in **error** or **warning** status, Clara provides you the list. Click on a server to get additional information about its status.

By selecting a server, Clara returns the current status of the server and, in case of error or warning, the issue description and the failing component(s). A direct link to HERO is available if you want to check the server in HERO and get additional details.

The screenshot shows the Clara interface with a header containing a profile icon, the name "Clara", and the subtitle "Your Automation Virtual Assistant". The main content area displays a server alert for IP 10.18.25.220 - BKM at 15:31. The alert details include: Date: 10/27/2020 11:20, HERO Environment: Test_HERO WA, Status: Error, Component: BKM, and Description: Broker process is running on 2 machines. Below the alert, Clara asks if the user wants to be notified when the server status changes. A dialog box with "Yes" and "No" buttons is shown, with "Yes" selected. The interface also shows a confirmation checkmark at 15:31.

Also, Clara asks if you want to be notified when the status of the server changes. Select **Yes** to receive Clara notification.



Monitoring a specific server

By selecting this action, you ask Clara to monitor a specific server on your behalf, and to send you a **notification** when the server status changes.

Clara
Your Automation Virtual Assistant

This is the list of MDM servers I found in HERO. Select one to monitor it.

PROD:

- 10.14.37.184 - Main
- 10.14.37.37
- 10.18.25.240 - Main
- 10.14.05.30

OTHER:

- 10.18.25.240 - Main
- 10.14.05.30

You can click on the MDM server name to get further details, otherwise you can ask for the BKM servers

Show me the BKM servers

Quit

15:30

Show me the BKM servers

15:31 ✓

Clara provides you with the full list of your MDM and BKM servers (both in PROD and OTHER type environments in HERO).

By selecting a server, Clara returns the current status of the server and, in case of error or warning, the issue description and the failing component(s). A direct link to HERO is available if you want to check the server in HERO and get additional details.

The screenshot shows the Clara interface. At the top left is the Clara logo and name, 'Clara Your Automation Virtual Assistant'. On the right are three vertical dots and a minus sign. The main content area shows a server alert for IP 10.18.25.220 - BKM at 15:31. The alert details are: Date: 10/27/2020 11:20, HERO Environment: Test_HERO WA, Status: Error, Component: BKM, and Description: Broker process is running on 2 machines. Below the alert is a confirmation dialog asking 'I can monitor this server for you and send you a notification when its status changes. Do you want to be notified?' with 'Yes' and 'No' buttons. The 'Yes' button is selected. At the bottom right, the word 'Yes' is displayed next to a vertical bar and a checkmark, with the time 15:31.

Also, Clara asks if you want to be notified when the status of the server changes. Select **Yes** to receive Clara notification.

Are there any issues with my servers?

By selecting this action, you ask Clara to quickly discover if there are any issues with your servers.

Get the status of all the PROD servers

Monitoring a specific server

Are there any issues with my servers?

You can click on the action name to get further details, otherwise just click

Back

Quit

15:30

Are there any issues with my servers?

15:31 ✓

To discover if there are any issues with WA servers

15:30

Do you want me to perform this task?

Yes

No

15:30

Yes


15:31 ✓

Good News! It seems there are no WA servers in error!

If Clara discovers servers in error or in warning status, she provides the list. A direct link to HERO is available if you want to check any server in HERO.

You can check the server directly in [HERO](#), otherwise select a server to get further details.

Here is the list of servers in error:

10.14.37.184 - Main	
10.14.36.50	

Here is the list of servers in warning:

10.14.37.182 - Backup	
10.14.36.22 - Server 2	

You can click on the server name to get further details, otherwise just click

[Quit](#)

15:30

10.14.37.184 - Main

15:31 ✓

By selecting a server, Clara returns the issue description, the failing component(s) and the environment type. A direct link to HERO is also available if you want to check the server in HERO and get additional information.

10.14.37.184 - Main

15:31 ✓

Server alerts on Main:

Date: 10/27/2020 13:20
HERO Environment: Test_HERO WA
Environment Type: PROD
Status: Error
Component: BKM
Description: Broker process is running on 2 machines

Date: 10/27/2020 15:20
HERO Environment: Test_HERO WA
Environment Type: PROD
Status: Error
Component: MDM
Description: Broker process is running on 2 machines

If you want, you can check the server directly in [HERO](#) and get additional information

15:30

Type a message...

Also, Clara asks if you want to be notified when the status of the server changes. Select **Yes** to receive Clara notification.

Clara helps in problem determination

Clara can be very helpful in troubleshooting the most common error situations for which the Workload Automation Support Team is usually involved. Clara provides also useful links to technical notes, articles, L3 documentation available about issues and known limitations.

From the *What can you do?* conversation flow, select **Troubleshooting**.

The screenshot shows a chat window with a list of categories. The 'Troubleshooting' category is highlighted in blue. Below the list, there is a message: 'You can click on the action name to get further details, otherwise just click'. The time '15:30' is shown. To the right, the word 'Troubleshooting' is displayed with a vertical orange bar and a checkmark, with the time '15:31' below it. Below this, another list of options is shown, with 'Open a ticket' highlighted in blue. A second message below it says: 'You can click on the action name to get further details, otherwise just click'.

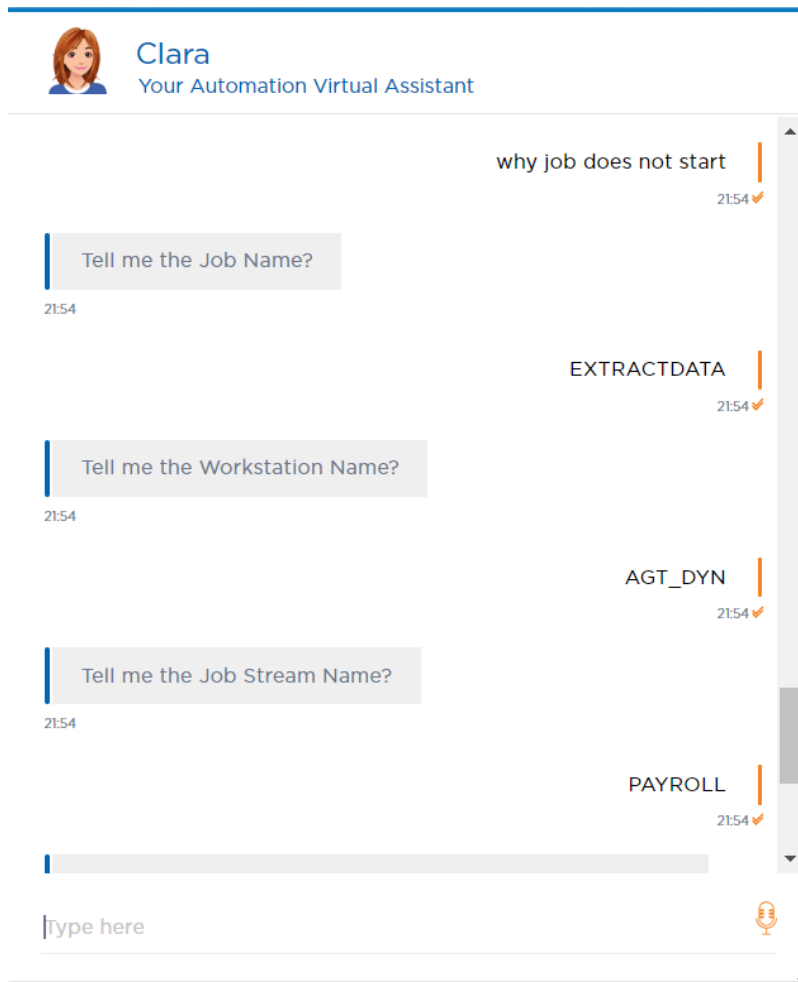
The following options are available:

- **Why a job doesn't start?** - You can understand why a job is not starting (available for WA only). See [Why a job doesn't start](#).
- **Are there any jobs in error?** - You can find any jobs in error in the daily plan (available for WA and WAz). See [Are there any jobs in error?](#)
- **Are there any jobs in late?** - You can find any jobs in late in the daily plan (available for WA and WAz). See [Are there any jobs in late?](#)

- **Open a ticket** - Clara provides you with a link to the Service Platform of your choice (available for WA and WAZ). For details, see [Open a ticket](#).

Why a job doesn't start?

You can ask Clara why a job doesn't start. If the job doesn't start because the job limit of the workstation is set to zero, Clara asks if you want to increase the limit. Also, Clara can recognize if the job doesn't start because the agent is unlinked, stopped, if there are dependencies, or the job priority is set to zero. In all these situations, Clara asks if she can run an action for you, such as changing the priority, increasing the limit, and so on.



The screenshot displays the Clara chat interface. At the top left, there is a profile icon of Clara, a woman with red hair, followed by the text "Clara" and "Your Automation Virtual Assistant". The chat history shows a sequence of user questions and Clara's responses:

- User: "why job does not start" (21:54) ✓
- Clara: "Tell me the Job Name?" (21:54)
- User: "Tell me the Workstation Name?" (21:54)
- Clara: "EXTRACTDATA" (21:54) ✓
- User: "Tell me the Job Stream Name?" (21:54)
- Clara: "AGT_DYN" (21:54) ✓
- Clara: "PAYROLL" (21:54) ✓

At the bottom of the chat window, there is a text input field with the placeholder text "[type here]" and a microphone icon to its right.

The screenshot shows a chat window for Clara, 'Your Automation Virtual Assistant'. The interface includes a header with Clara's name and a profile picture. The chat history shows a user asking for a job stream name, Clara responding with 'PAYROLL', and the user providing a specific job ID. Clara then lists two job IDs and asks for clarification. The user provides another job ID, and Clara responds with an error message about a workstation not being started. At the bottom, there is a text input field with a microphone icon.

Clara
Your Automation Virtual Assistant

21:54 ✓

Tell me the Job Stream Name?

21:54

PAYROLL

21:54 ✓

I found more than one jobs with the provided data, the jobs found are the following:

AGT_DYN#PAYROLL(2018-07-04T07:02:00).EXTRACTDATA

AGT_DYN#PAYROLL(2018-07-04T09:38:00).EXTRACTDATA

Please tell me wich one of these are the job you where looking for

21:54

AGT_DYN#PAYROLL(2018-07-04T09:38:00).EXTRACTDATA

21:54 ✓

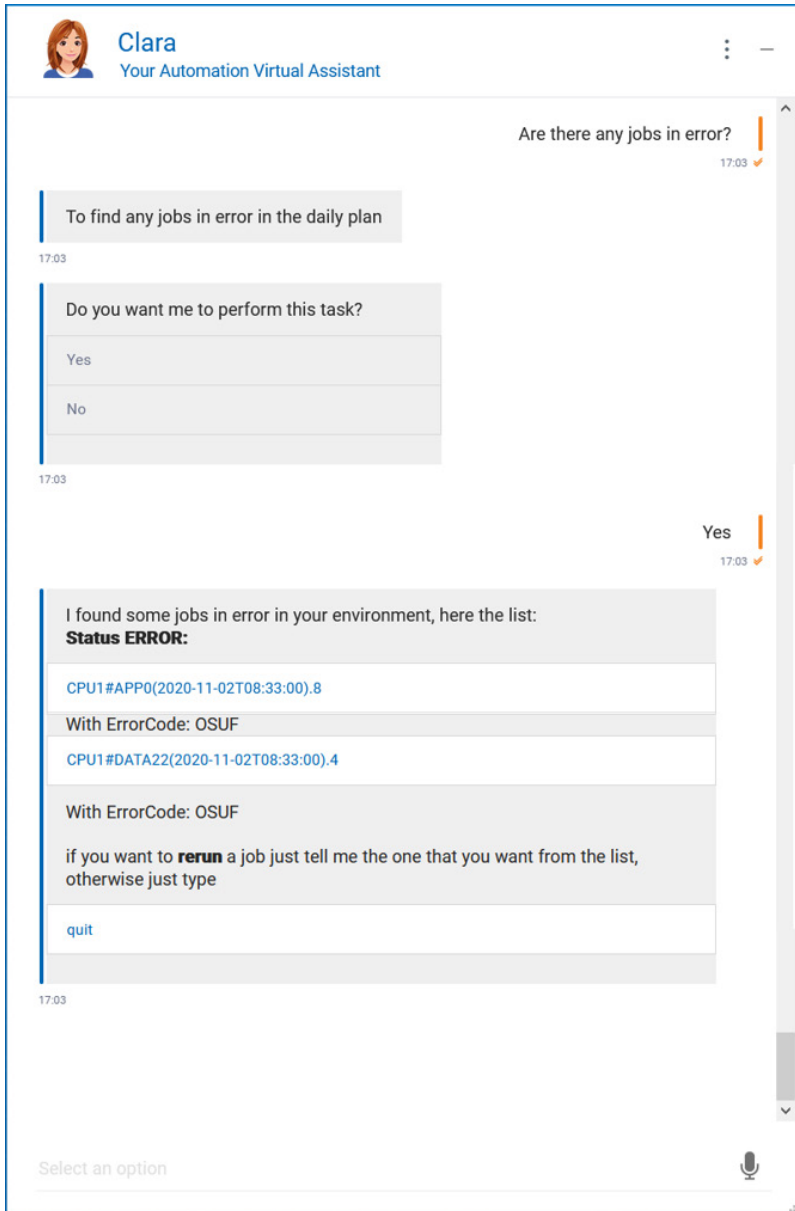
AWSJCO199I The workstation"AGT_DYN" is not started. Start the workstation to run jobs in READY status.

21:54

|type here

Are there any jobs in error?

You can ask Clara if there are jobs in error. Clara will list all the jobs in error and ask you if you want to any.



Are there any jobs in late?

You can ask Clara whether there are jobs in late. Clara will lists all the jobs in **High Risk** or **Potential Risk**.



Clara
Your Automation Virtual Assistant

are there any jobs in late?

21:46 ✓

Ok. Do you find any of the options below relevant to your query? If yes, select an option to help me understand your query better:

1 Show high risk critical jobs

2 What actions are taken for late jobs or job streams?

3 What is bm late every option?

4 None of the above

If none of the above is relevant, type "None" to see more options

21:46

1 Show high risk critical jobs

21:46 ✓

I've found more than one critical job with a **Potential** or **High** risk of not matching the deadline. The list of jobs is

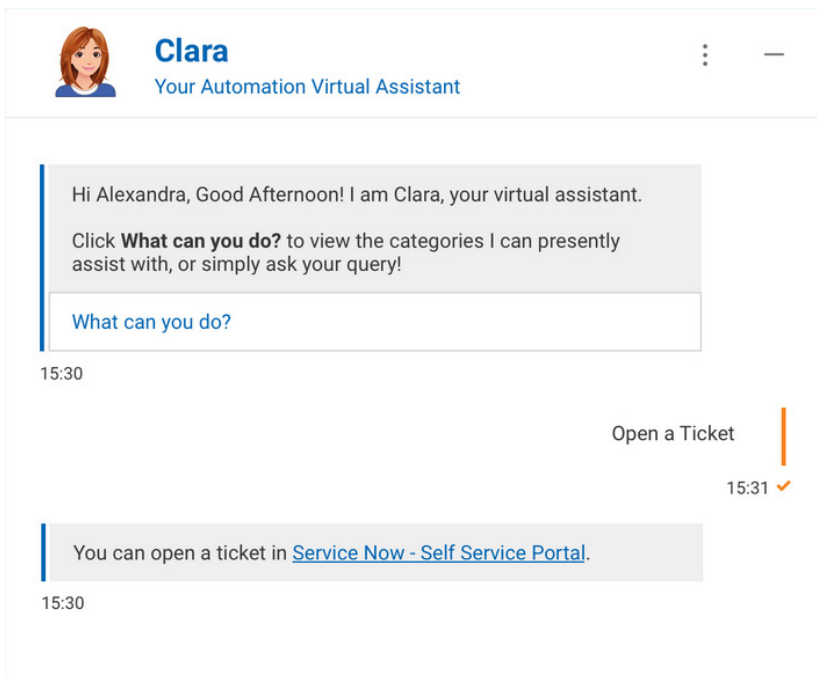
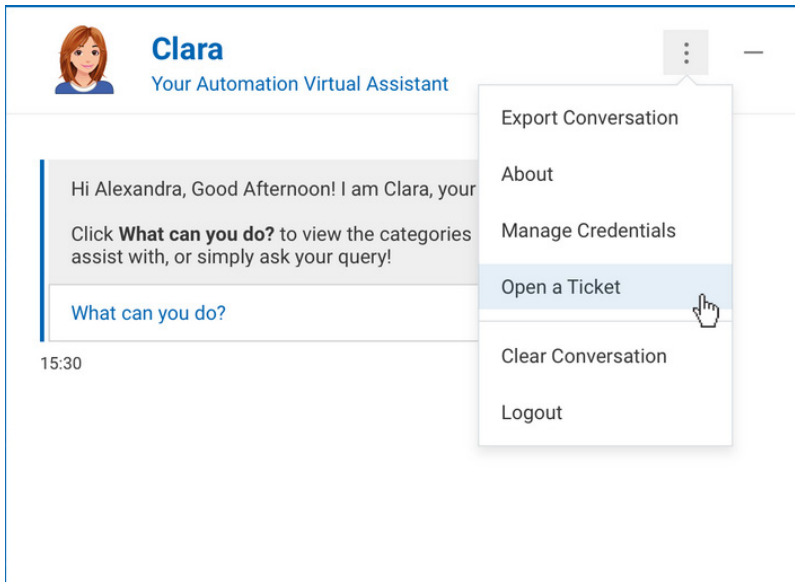
|type here



The screenshot shows the Clara chat interface. At the top left, there is a profile icon for Clara, labeled "Clara Your Automation Virtual Assistant". The main chat area displays a message from Clara: "1 Show high risk critical jobs" with a timestamp of 21:46 and a checkmark. The message content is: "I've found more than one critical job with a **Potential** or **High** risk of not matching the deadline. The list of jobs is the following: **HIGH RISK:** AGT_DYN#PAYROLL(2018-07-04T09:38:00).ANALYZEDATA". Below this, it says "More info: the job is in late status, estimated end: 2018-07-04T19:45:02, actual end: undefined" and "If you want to know which jobs are causing the delay, select one of the jobs listed above, otherwise type" followed by a "quit" button. A second message from Clara follows, with the same timestamp and checkmark, containing the text: "Here the list of all the jobs that are causing the delay: AGT_DYN#PAYROLL(2018-07-04T09:38:00).EXTRACTDATA" and "More info: the job is in late status, estimated end: 2018-07-04T19:45:51, actual end: undefined". At the bottom, there is a text input field with the placeholder "Type here" and a microphone icon.

Open a ticket

When you select this action, Clara provides you with a link to the Service Platform of your choice. You can open the same link by clicking **Open a ticket** in the menu on the upper right corner of Clara window.



If you didn't specify your preferred Service Platform in your WA environment configuration file, Clara will inform you with a message.

For details about how to specify your Service Platform in your WA environment configuration file, see: [Installing and Configuring Clara](#).

Troubleshooting Guide

About the Troubleshooting Guide

The Troubleshooting Guide provides information about how to collect logs and resolve problems in HCL Clara.

[Collecting logs](#)

[Troubleshooting Clara](#)

Collecting logs

Log files for Clara components are located inside the respective containers.

In case of issues, contact Clara Support Team and provide a .zip file containing all Clara log files.

To extract Clara log files and create a .zip file, run the following procedure:

1. For each container, extract the log file with the following commands:
 - a. For reverse proxy container:


```
$(docker logs clara-nginx -t) > clara-nginx.log
```
 - b. For identity management container:


```
$(docker logs clara-keycloak -t) > clara-keycloak.log
```
 - c. For credential manager container:


```
$(docker logs clara-credmanager -t) > clara-credmanager.log
```
 - d. For AutoBot engine container:


```
$(docker logs clara-autobot -t) > clara-autobot.log
```
 - e. For Kibana container:


```
$(docker logs clara-dashboard -t) > clara-dashboard.log
```
 - f. For extension engine container (extension engine provides NLP functionality to Autobot):


```
$(docker logs clara-extension -t) > clara-extension.log
```
 - g. For Elasticsearch container:


```
$(docker logs clara-datastore -t) > clara-datastore.log
```
 - h. For Apache Solr container:


```
$(docker logs clara-search-engine -t) > clara-search-engine.log
```
 - i. For Clara search engine container (Clara search engine is used to configure Solr cores at installation time):


```
$(docker logs clara-search-config -t) > clara-search-config.log
```
 - j. For microservice container (microservice handles semantic search queries):


```
$(docker logs clara-queryservice -t) > clara-queryservice.log
```
 - k. For Clara asynchronous service container (Clara asynchronous service handles the execution of asynchronous task interacting with WA):


```
$(docker logs clara-asynservice -t) > clara-asynservice.log
```
 - l. For Clara rabbitmq container (Clara rabbitmq handles message exchange between front-end and asynchronous service):

```
$(docker logs clara-rabbitmq -t) > clara-rabbitmq.log
```

For your convenience, you can run the above commands all together. Copy and paste the following expression:

```
$(docker logs clara-nginx -t) > clara-nginx.log ; $(docker logs clara-keycloak -t) > clara-keycloak.log ;  
$(docker logs clara-credmanager -t) > clara-credmanager.log ; $(docker logs clara-autobot -t) > clara-  
autobot.log ; $(docker logs clara-dashboard -t) > clara-dashboard.log ; $(docker logs clara-extension -t) >  
clara-extension.log ; $(docker logs clara-datastore -t) > clara-datastore.log ; $(docker logs clara-search-  
engine -t) > clara-search-engine.log ; $(docker logs clara-search-config -t) > clara-search-config.log ;  
$(docker logs clara-queryservice -t) > clara-queryservice.log ; $(docker logs clara-asynservice -t) >  
clara-asynservice.log ; $(docker logs clara-rabbitmq -t) > clara-rabbitmq.log
```

2. Create a .zip file by running the following command:

```
tar -czvf clara-logs.tar.gz ./*.log
```

Troubleshooting Clara

Symptom

Docker login returns: "Error response from daemon: login attempt to https://854793243809.dkr.ecr.us-east-1.amazonaws.com/v2/ failed with status: 403 Forbidden"

Cause and solution

Your credentials are not valid. Request new credentials from your vendor.

Symptom

Docker or docker-compose commands return: "Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.38/containers/json: dial unix /var/run/docker.sock: connect: permission denied"

Cause and solution

You don't have the right permissions for Docker. Use commands with root permissions ("sudo" or "sudo su") or refer to [Docker documentation](#).

Symptom

Nginx container doesn't start returning the following error: "ERROR: for nginx Cannot start service nginx: driver failed programming external connectivity on endpoint ngiinx".

Cause and solution

Stop and start docker compose by running the following commands:

1. docker-compose stop
2. docker-compose up

If the problem persists, run the following steps:

1. restart docker with **docker restart** command. See [Docker command usage](#).
2. docker-compose up

Symptom

When uploading the Autobot License file, it returns: "Invalid Licence, Bot Framework Not Initialized".

Cause and solution

Your license file is not valid, Request a new license file from your vendor.

Symptom

During Clara installation, the command **clara.sh --load-project <prj>** fails with the following error: { "status": "Exception occurred:java.net.UnknownHostException: datastore"}.

Cause and solution

The most common cause for this error is a network issue within "clara-net" (docker network ls). In this situation the containers are unable to communicate with each other.

To verify if you are in this situation, run the following steps:

1. Connect to clara-autobot container:

```
docker exec -ti clara-autobot bash
```

2. Run the command:

```
curl http://datastore:9200
```

if you get something like the following output, the connection is working:

```
{
  "name": "Vtmbdmn",
  "cluster_name": "autobot",
  "cluster_uuid": "b3r4-1lkR6OHk49dnpd3gA",
  "version": {
    "number": "5.6.2",
    "build_hash": "57e20f3",
    "build_date": "2017-09-23T13:16:45.703Z",
    "build_snapshot": false,
    "lucene_version": "6.6.1"
  },
  "tagline": "You Know, for Search"
}
```

If you don't get any answer to the curl command, run the following steps:

1. Stop all containers:

```
./clara.sh --stop
```

2. Stop the docker daemon:

```
systemctl stop docker (the command depends on platform)
```

3. Start docker and Clara:

```
systemctl start docker
```

```
./clara.sh --start
```

Symptom

Accessing Clara website you get a security warning: " Your connection is not private".

Cause and solution

Add an exception to trust the website

Symptom

Clara is installed on the same machine of Workload Automation. WA is installed on docker but Clara cannot contact WA

Cause and solution

Link Clara container with the WA network, by running the following command:

docker network connect docker_wa-net clara-autobot

and use "wa-server" as dns of WA engine.

Symptom

ElasticSearch does not start, and the log shows the following error:

```
"2018-09-21T14:11:16,039][INFO ][p.e.b.BootstrapChecks ] [qVAFkOU] bound or publishing to a non-loopback address, enforcing bootstrap checks
```

```
ERROR: [2] bootstrap checks failed
```

```
[1]: max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]
```

```
[2]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]"
```

Cause and solution

To solve this problem, run the following procedure.

How to verify and set ulimit parameter

1. Check the maximum number of open files for the current user (the user that started the Docker daemon) by running the command **ulimit -n**
2. Verify that the number of allowed open files for the current user is at least 65536
3. Check the Hard limit for the current user, by running the command **ulimit -n -H**
4. Check the Soft limit for the current user, by running the command **ulimit -n -S**
5. In case the value of Hard or Soft limit is lower than 65536, increase its value **permanently**, by editing the file:

```
/etc/security/limits.conf
```

```
[domain] [type] [item] [value]
```

where:

- [domain] can be a username, a group name, or a wildcard entry
- [type] is the type of the limit and can have the following values:
 - soft: a soft limit which can be changed by user
 - hard: a cap on soft limit set by super user and enforced by kernel
- [item] is the resource for which you are setting the limit

For example, for a user with id hmuser run the following steps:

1. Add or modify soft and hard limits as follow:
 - hmuser soft nofile 65536
 - hmuser hard nofile 65536
 2. Activate the new values by running the following command **sysctl -p**
 3. Update the following files:
 - /etc/systemd/user.conf
 - /etc/systemd/system.conf
- by adding the following line:
- ```
DefaultLimitNOFILE=65536
```
4. Login again with user hmuser and verify the new limits before starting any process

### Symptom

The log shows the following error: "[1]: max virtual memory areas vm.max\_map\_count [65530] is too low, increase to at least [262144]".

### Cause and solution

On Linux, run the following procedure.

#### How to verify and set the available virtual memory

1. To verify the available virtual memory, run the following command as the user that started the Docker daemon:

```
sysctl vm.max_map_count
```
2. If the command output shows a value lower than 262144, run the command:

```
sysctl -w vm.max_map_count=262144
```
3. To set this value permanently, edit the vm.max\_map\_count setting in /etc/sysctl.conf.
4. Add the following as last row, or edit the row if present:

```
vm.max_map_count=262144
```
5. Verify the new value after reboot.

### Symptom

Keycloak container restarts, or the log shows the following error:

"Oops...I understand that you are trying to access the intent .. But I am not trained/configured to take any action on this yet. Please contact administrator to get this added.

Security-Enhanced Linux (SELinux) on the hypervisor must be disabled or the permissions must be set up correctly".

### Cause and solution

To set SELinux to permissive, run the following procedure.

#### How to set SELinux to permissive

1. Run the following commands:

```
sed -i s/^SELINUX=.*$/SELINUX=permissive/ /etc/selinux/config
```

```
setenforce 0
```

```
sed -i s/^SELINUX=.*$/SELINUX=disabled/ /etc/selinux/config
```

2. Restart the system to save the changes permanently.
3. After you restart the system, you can use the `getenforce` command to check the SELinux status.

### Symptom

On first installation, Clara returns "Server internal error".

### Cause and solution

Clara needs to be reinstalled:

```
./clara.sh --down --vol
```

```
./clara.sh --up
```

```
./clara.sh --load-project HWA --license ./license.properties
```

### Symptom

On first installation, the installation script returns the following error message:

"The image for the service you're trying to recreate has been removed. If you continue, volume data could be lost. Consider backing up your data before continuing.

Continue with the new image Y|N?"

### Cause and solution

You must first load Clara images from the `clara_images.tar.gz` file. This file is in the same path as the installation script.

To load Clara images, run the installation script with the appropriate options:

```
./clara.sh --load-images
```

### Symptom

After upgrading Clara from previous version, one or more of the following messages appear when logging into Clara:

- "Failed to initialize the authentication adapter"
- "Undefined"

### Cause and solution

You might have missed a step of Clara [upgrade procedure](#).

Run the command `./clara.sh --migrate-data` to complete the upgrade procedure and then reload Clara UI page.

### Symptom

At login, the following message appears on Clara's UI: "response indicates failure, status=400, body={\"error\": \"unauthorized\_client\", \"error\_description\": \"Invalid client secret\"}"

### Cause and solution

You might have specified an invalid key (CLIENT\_SECRET) in the .env configuration file. Verify that the key specified in the .env file is equal to the one specified in keycloak. If the key is not the same, run the following procedure:

1. From Keycloak console, in the Credentials tab, copy the content of the **Secret** field
2. In the .env file (path <BUILD\_DIR>/clara/.env), paste the valid key into the CLIENT\_SECRET parameter, overriding the current value
3. From the <BUILD\_DIR> directory, run the commands `./clara.sh --up -nc`
4. Reload Clara UI page