# ***HCL Clara V2.0.0.0***

# Contents

# 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 enhances **HCL Workload Automation** and **HCL Workload Automation for Z** customer experience with Natural Language Processing (NLP) interactions, enabling customized self-service automation with immediate 24x7 response. HCL Clara is the product expert that users can consult to learn the product capabilities, execute routine tasks and get troubleshooting assistance.

In Clara V2,  a completely redesigned User Interface improves customer interaction, usability and accessibility. A more efficient entity recognition allows for faster conversation flow. A brand new dashboard makes Clara customization easier and provides useful metrics.

For details about Clara V2 enhancements, see What's New.

For information about **HCL Workload Automation** and **HCL Workload Automation for Z**, see HCL Workload Automation Documentation.

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

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 Schedulers, manages the how-to questions and initiates the 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 user interface to scheduling activities, 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 Dashboard
- Actions and Runbooks implementation leveraging product APIs
    - Clara leverages Workload Automation 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 jobs or job streams
- Re-run jobs

Troubleshoot Workloads

- Why the job is not starting?
- Why did the job fail?

# What's New

See what's new with Clara releases.

## Clara V2.0

**Clara V2.0 includes the following enhancements:**

1. Support of Workload Automation V10.1.0 for Distributed and Z systems.
2. Clara is based on Rasa, the new state-of-the-art on-prem framework for natural language understanding, dialogue management, and integrations.
3. Clara UI has been completely redesigned to improve user interaction.
4. Clara new avatar showing emotions for empathic communication.
5. Chat bubbles, colors and input boxes to enhance usability and accessibility.
6. Smart cards to handle action categories.
7. Tips for next action.
8. Faster and more friendly conversation flow.
9. Better recognition of voice commands through Web Speech APIs.
10. Improved context awareness and entity recognition in the current Workload Automation environment.
11. New dashboard for configuration and metrics.
12. Multilanguage enablement.
13. The justification for auditing purposes is no longer needed. It has been replaced by an internal audit mechanism that tracks all the actions in the audit store as "executed by Clara".
14. Support of **Slack** and **Teams** messaging tools.
15. Availability on HCL Software Factory (SoFy), a catalog of Kubernetes enabled products that can be deployed to a cloud-native environment. This functionality is released as **technical preview**.

## Clara V1.0.0.7

**Clara V1.0.0.7 includes the following enhancements:**

1. Support of Workload Automation V10.1.0 for distributed and z systems.

2. Clara can be integrated in the Dynamic Workload Console. For the steps to run on the Dynamic Workload Console, see the following topics in the Workload Automation documentation:
    - *Integrating Clara in the Dynamic Workload Console*
    - *Integrating Clara after an environment update*

## Clara V1.0.0.6

**Clara V1.0.0.6 includes the following enhancements:**

1. Support of WA V9.5 FP5.
2. Availability on HCL Software Factory (SoFy), a catalog of Kubernetes enabled products that can be deployed to a cloud-native environment. This functionality is released as **technical preview**.
3. Multilanguage enablement. See: Configuring language translation and Selecting a conversation language. This functionality is released as **technical preview**.
4. The new HCL Translator service component that enables language localization. See: Installing and configuring HCL Translator. This functionality is released as **technical preview**.
5. Clara V1.0.0.6 no longer requires Autobot license.
6. Clara UI has been completely renewed to improve the user experience.

## 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. Wronk 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, 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**

# Product Videos

You can find videos demonstrating Clara capabilities and benefits in videos published on the HCL Software YouTube channel.

# Customer Support

To contact HCL Customer Support or create a product case, see: Customer Portal.

# About the Installation and Configuration Guide

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

System Requirements

Installing and Configuring Clara

Clara availability on SoFy (tech preview)

Installing and Configuring HCL Translator

Upgrading Clara

Configuring Security

Accessing Clara Dashboard

Managing the intents

Managing user credentials

Managing the Workload Automation environments

Customizing Clara settings

Appendix

# System Requirements

Clara is installed using Docker.
You can install Clara on Linux 64 bit operating system.
Clara V2 is available also on HCL Software Factory (SoFy), a catalog of Kubernetes enabled products that can be deployed to a cloud-native environment. For Clara on SoFy System Requirements , see HCL SoFy Guides.

## 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 1.29.0+
- 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 9.5 fix pack 4**.

# Installing and Configuring Clara

Clara can be installed with Docker.

## 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.
5. Obtain your Clara license entitlement (**License Id**) through the **HCL Cloud License Server** (a working internet connection is required). For details, see What is the HCL Software License & Download Portal. The license entitlement and expiration date depend on the type of license you have purchased (if product, bundle, or trial). The License Id must be specified during the installation procedure.

## Installation procedure

To install and configure Clara, run the following procedure.

1. From HCL License Portal download the appropriate Clara installation package:
   - HCL CLARA for HCL Workload Automation V2.0.0.0

   Clara is available with 2 different types of installation packages:

   - Installation package for **offline installation**:
     HCL_CLARA_2.0.0.0_OFFLINE.tar.gz containing:
     - clara.tar.gz
     - HCL_CLARA_IMAGES.tar.gz
     - install.sh

- Installation package for **online installation**:
  HCL_CLARA_2.0.0.0_ONLINE.tar.gz containing:
  - clara.tar.gz
  - install.sh

  The **online** installation requires a working internet connection to reach the Entitled Registry that contains Clara images. Contact your HCL sales representative for the login details required to access the Entitled Registry. Execute the following command to log in into the Entitled Registry:
  **docker login -u <your_username> -p <your_entitled_key> hclcr.io**

  Clara images are as follows:

  - hclcr.io/solutions/clara/clara-manager:2.0.0.0
  - hclcr.io/solutions/clara/clara-rasa:2.0.0.0
  - hclcr.io/solutions/clara/clara-search-engine:2.0.0.0
  - hclcr.io/solutions/clara/clara-nginx:2.0.0.0
  - hclcr.io/solutions/clara/clara-keycloak:2.0.0.0
  - hclcr.io/solutions/clara/clara-asyncservice :2.0.0.0
  - hclcr.io/solutions/clara/clara-rabbitmq :2.0.0.0
  - hclcr.io/solutions/clara/clara-datastore :2.0.0.0

2. For the selected installation package (offline or online), extract the content of the tar.gz file(s) 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. You can properly modify the **<BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE** configuration file to set some common properties and customize Clara integration with **Slack** and **Teams** messaging tools. See below the sections about Customizing environment files and Integration with Slack and Teams.
   **Note:**
   After Clara installation, you can change or complete Clara customization by using **Clara Dashboard**. For details, see Clara Dashboard.
4. Open a bash shell and get ready to install Clara.
5. From the <BUILD_DIR> directory, run the command:

   **./install.sh [--load-images] --install [--install-path <*install_path*>] --accept-license [--port <*port_number*>] [--img-repo <*img_repo*>] --license-id <*license_id*>**

where:

**--load-images** is for offline installation only. Its is required to load Clara images into your environment.
**--install-path <*install_path*>** is optional if you wan to change the installation directory of your choice (default value is /opt/hcl).
**--accept-license** is required to accept the HCL Master License Agreement.
**--port <*port_number*>** is optional if you want to change the HTTPS port to use with Clara (default port is 443).
**--img-repo <*img_repo*>** is for online installation only. It is optional if you want to specify an Entitled Registry prefix different from the default prefix (hclcr.io/solutions/clara/.
**--license-id <*license_id*>** is required to set Clara License Id.

By default, before starting the installation, Clara checks the installation prerequisites. If you want to skip this check, add the --noprereq parameter. For details, see the install.sh command usage.

6. The installation script checks the prerequisites, 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 link **https://<IP:PORT>**
- Clara Dashboard at the link **https://<IP:PORT>/dashboard**

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).

Clara Dashboard can also be accessed from the drop-down menu of Clara chat window.

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 the Workload Automation environments that you want to manage with Clara and identify the current environment.

Go to **Clara Dashboard > Environments**. For details, see Environments.

- Define credentials for the product environments you want to manage with Clara. The **Credential Manager** component, accessible from Clara Dashboard, helps you map your product environment credentials with your Clara credentials (each Clara user can only manage his own product environment 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**

You can also reach the Keycloak administration console from Clara Dashboard, by clicking **Manage roles** in the Account icon drop-down menu on the dashboard header. To change the Keycloak default password or the default SSL certificates, see Configuring Security.

If you want to enable **multiple languages** with Clara through HCL Translator, see Installing and configuring HCL Translator.

After Clara installation, use the **Dashboard** to complete the following Clara settings:

- User credentials
- Clara intents
- Dashboard language
- Clara integration in the Dynamic Workload Console
- Dashboard Time Zone
- Workload Automation environments that you want to manage with Clara
- Cache synchronization
- SMTP server for email notification
- Translation services
- HERO integration

For details, see Clara Dashboard.

To start/stop Clara containers and services, see clara.sh command options.

## Integration with Slack and Teams

To integrate Clara with **Slack** and **Teams** messaging tools, you must run the following steps:

1. Install Clara on a machine having a public internet connection.
2. Provide the required Slack and Teams parameters in the **<BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE** configuration file.
3. Ensure that the **email id** that you set in **Clara Dashboard > Credential Manager** for your Clara user, matches the email id you set on Slack and Teams.
4. Run some configuration steps for the messaging tools:

    - **Slack**
        1. Define Clara app on Slack by following the instructions contained in the Slack documentation. You can also watch a useful tutorial video.
        2. In addition to the settings listed in the Slack documentation, set the following parameters:

            **Additional bot events:**
            app_home_opened

            **Additional scopes:**
            users.profile:read
            users:read.email

            **Interactivity & Shortcuts -> Request URL:**
            https://<clara_url>/webhooks/slack_auth/webhook

            **Event Subscriptions -> Request URL:**
            https://<clara_url>/webhooks/slack_auth/webhook

        3. On Slack, go to **General Information -> Display Information** and set the fields as indicated in the **App settings** table below.

    - **Teams**
        1. Register Clara's bot with Azure by following the instructions contained in the Azure documentation. You can also watch a useful tutorial video.
        2. Define Clara app on Teams. A tutorial video for this step is also available.
        3. On Teams, set the parameter **Configuration -> Messaging endpoint:** https:///<clara_url>/webhooks/botframework_auth/webhook.
        4. On Teams, go to **App Details -> Descriptions** to set the name, and the short and long descriptions for the app, as indicated in the **App settings** table below.

5. Go to **App Details -> Branding** to set the app icon as indicated in the **App settings** table below.

| App settings | |
|---|---|
| **App name** | Clara |
| **Short description** | Your Workload Automation Virtual Assistance |
| **Long description** | Clara is an Intelligent Virtual Assistant that enhances HCL Workload Automation customer experience with Natural Language Processing (NLP) interactions, enabling customized self-service automation with immediate 24x7 response. HCL Clara is the product expert that users can consult to learn the product capabilities, execute routine tasks and get troubleshooting assistance. |
| **App icon for Teams** | Download Teams_icon.zip |
| **App icon for Slack** | Download Slack_icon.zip |
| **Background color for Teams** | #71D9F3 |
| **Background color for Slack** | #01539B |

You can follow your conversation with Clara even using both channels. For example, you can ask your question to Clara on Slack, and then continue the conversation on Teams.

## Customizing environment files

Both for Workload Automation and Workload Automation for Z**,** before starting Clara installation, modify the **<BUILD_DIR>/clara/config/Clara-WA.env.TEMPLATE**

configuration file to set Clara common properties and optionally configure Clara integration with Slack and Teams messaging tools.

| Common properties | |
|---|---|
| NGINX_PORT | TCP/IP Port to use with Clara [Default: 443] |
| NGINX_CERTS | Path where nginx certificates for Clara are stored [Default: ./certs/nginx] |
| **Properties for Clara integration with Messaging tools (Slack and Teams)** | |
| BOTFRAMEWORK_APP_ID | ID for Clara Teams account |
| BOTFRAMEWORK_APP_PASSWORD | Password for Clara Teams account |
| SLACK_TOKEN | The token generated by the Slack bot |
| SLACK_CHANNEL | The Slack bot channel ID |
| SLACK_SIGNING_SECRET | The Slack unique string for Clara. |
| **Properties for source docker images** | |
| IMG_PREFIX | Clara docker image repo prefix when images are not available locally. [Default: hclcr.io/solutions/clara/] |
| TAG | Tag for Clara docker images based on Clara version, optional. [Default: 2.0.0.0] |

**After Clara installation**, if a re-configuration is needed to integrate Clara with Slack or Teams messaging tools, you must modify the .env file created during the installation process and located in the folder <BUILD_DIR>/clara. Then, run the following command to activate the updates:
**clara.sh --up --nc**

## install.sh command

install.sh [options]

**OPTIONS:**

| | |
|---|---|
| -h | Show this message |
| -c, --prereq | Prerequisites (skip any other option if preser |
| --noprereq | Pass this parameter to skip all OS prereq ch |
| --load-images | Load Clara images from HCL_CLARA_IMA |
| --install [--install-path <*install_path*>] | Install Clara<br>Add --install-path <path> to change path fr |
| --accept-license | Pass this parameter to accept the HCL Mas |
| --port <*port_number*> | Pass this parameter to overwrite the HTTP<br>Default HTTPS port for Clara is 443. |
| --img-repo <*img_repo*> | Set source repo prefix for Clara images whe<br>locally. Example: --img-repo example.cor |
| --license-id <*license_id*> | Set Clara license ID. Example: --license-id |

Steps:

1. Check Clara prerequisites. Run the command:
   ./install.sh -c

2. Load Clara images (run only if you need to import new images for an offline installatio
   ./install.sh --load-images

3. Install Clara. Run the command:
   ./install.sh --install [--install-path /opt/hcl] --accept-license --noprereq

## 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

--down [--vol] Stop and remove Clara containers and network
Specify --vol to remove also persistent volumes

--start Start Clara services

--stop Stop Clara services

--restart Restart (stop and start) Clara containers

--join-net Link Clara network to WA network when they are both running

--remove Print commands to clean-up Clara containers, persistent volu

--print-urls Print Clara URLS

*Examples:*

*First Clara start-up:*
*./clara.sh --up*

*Stop Clara containers:*

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

# Clara availability on SoFy

Clara V2 is available on HCL Software Factory (SoFy), a catalog of Kubernetes enabled products that can be deployed to a cloud-native environment.

SoFy uses Docker images and Helm technology to deploy Clara to a Kubernetes cluster of your choice (public or private). HCL SoFy provides a temporary environment-sandbox to deploy and test solutions. When you are ready to implement the solution, simply download the Helm installation package and run the software on the cloud provider of your choice. SoFy solutions are portable across all Kubernetes environments. You can also take advantage of SoFy's unique Kubernetes monitoring and administration tools.

For complete details about deploying services and products enabled for Kubernetes as docker images and helm charts, see HCL SoFy Guides. Search for Clara in the HCL SoFy Catalog.

# Installing and Configuring HCL Translator

Clara V2 provides **multilanguage enablement** through any the following services:

- HCL Translator (default)
- Google Translation Service
- Azure Translation Service

**Note:**
Multilanguage enablement in Clara is obtained through machine translation. Therefore, it could sometimes produce inaccurate results.
If you are using HCL Translator and you find that the translation is not accurate, you can submit your feedback through the **HCL Software portal** by providing the original sentence and the correct translation.
In this way, the HCL Translator development team can use your input to re-train the language model and improve the translation quality.

**HCL Translator** is an API based service. It is based on Machine Learning models that have been trained and fine-tuned to understand context-specific terms or phrases used in the Workload Automation domain, thus greatly improving the user experience. It supports bidirectional automatic translation from English to French, German, Italian, Portuguese, and Spanish.

After Clara installation, you can customize multilanguage enablement in Clara, by selecting one of the available services. For details, see Translations.

To use HCL Translator, you must first install it. HCL Translator is available on HCL Software Factory (SoFy), a catalog of Kubernetes enabled products that can be deployed to a cloud-native environment.
It is also available, together with Clara, on HCL License Portal as a package for on-premises installation with Docker.
To deploy HCL Translator, run the following procedure.

## Installation

1. Download the Translator package from the source repository and extract it with a tool available on your system.
2. Open a bash shell and move to the extracted directory with the following command: **cd <extract-dir>**

where <extract-dir> is the path where you have extracted the package.

3. Load the Docker images by running the following command:
   **./translator.sh load**
4. Install the Translator in a directory of your choice by running the following command:
   **./translator.sh install --install-dir=<install-dir>**
   where <install-dir> is the installation directory of your choice (default value is /opt/hcl/).
5. Download the language packages of your interest from the source repository and move them to <install-dir> without extracting them.
6. On the bash shell move to the installation directory with the following command:
   **cd <install-dir>**
7. Install the language packages by running the following command:
   **./translator.sh install-language --language-file=<lang-package>**
   where <lang_package> is the name of the language package file (e.g. `it.tar.gz`).
   **Note:** you can delete the package file after it has been successfully installed.
8. Repeat the previous step for each language that you have downloaded.
9. Configure the Translator options by following the Configuration section.
10. Run the Translator with the following command:
    **./translator.sh up**

## Configuration

To configure the Translator, open the <install-dir> and edit the **.env** file with a text editor of your choice:
This file contains the environment variables used by HCL Translator in the from `VARIABLE=VALUE`. You can edit the `VALUE` based on your needs.
Here's the meaning of each variable:
**INSTALL_DIR** is the directory where HCL Translator is installed. The default value is `./` that represents the current working directory. You should not change this if you followed the standard installation procedure.
**IMG_REGISTRY** is the registry of the Docker images. You should not need to change this variable.
**IMG_TAG** is the tag of the Docker images that represents the Translator version. You should not need to change this variable.
**NETWORK_NAME** is the Docker network the translator must connect to. The default is `clara-net`, so the Translator uses the network of Clara and is not accessible from the outside. Change this only if you install the Translator on a different machine than Clara's.
**EXTERNAL_NETWORK**, if *true* (default), the Translator uses a pre-existing Docker network. If *false*, it creates a new one. If you want to use `clara-net`, you must leave the

default value. Set this to *false* only if you install the Translator on a different machine than Clara's.

**docker-compose.yaml**

This file contains the configuration of Docker Compose. You should not edit this file unless you install the Translator on a different machine than Clara's. In this case, you should uncomment the following lines by removing the `#` at the beginning:

```
# ports:
#     - <PORT>:5000
```

In this way, the Translator will be exposed to the outside. Replace <PORT> with the desired port that you want to expose.
WARNING: exposing the Translator to the outside is a potential security issue, because it is not protected by any authentication. So, be sure that the exposed port is only accessible by the Clara machine.

### translator.sh script

**Usage**: ./translator.sh COMMAND [OPTIONS]

This script manages the translator deployment.

**Commands**:

| | |
|---|---|
| load | Loads the container images (required before the first start) |
| install | Installs translator in a directory (by default in /opt/hcl/translator) |
| install-language | Installs a language in a directory (by default the current working direct |
| up | Creates the containers and starts the application |
| stop | Stops the containers |
| restart | Restarts the containers |
| down | Removes the containers (but not volumes) |
| down-volumes | Removes the containers and volumes |

**Options**:

| | |
|---|---|
| --images | The archive path of the translator images (defaults to ./translator-image |
| --install-file | The archive path of the translator base files to install (defaults to ./translat |
| --install-dir | The directory where translator must be installed (defaults to /opt/hcl/trans |
| --language-file | The archive path of the language file to install (defaults to ./it.tar.gz) |
| --language-dir | The directory where the language must be installed (defaults to the cu |

# Upgrading Clara

You can upgrade your Clara installation from V1.0.0.6 or above to V2.0.0.0.

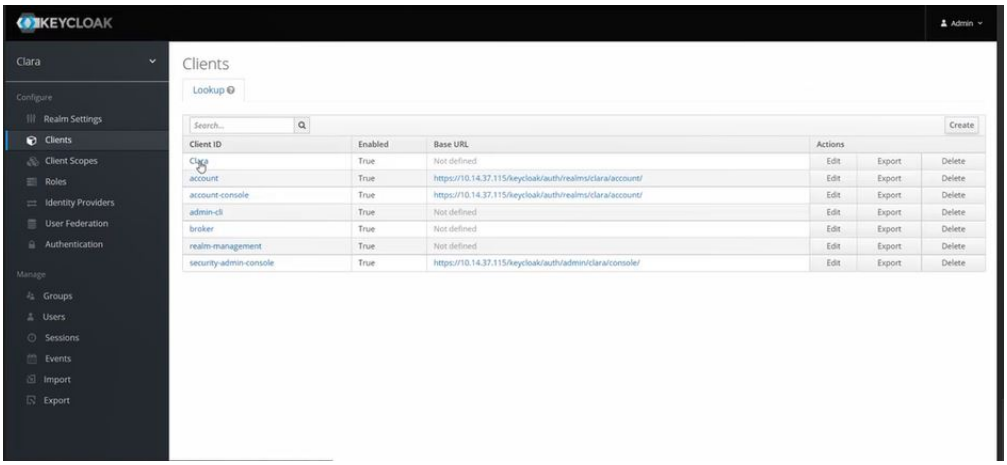
Before upgrading Clara, if you have modified the Knowledge Base of your previous installation by adding custom FAQs and you want to export them, run the following steps:

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

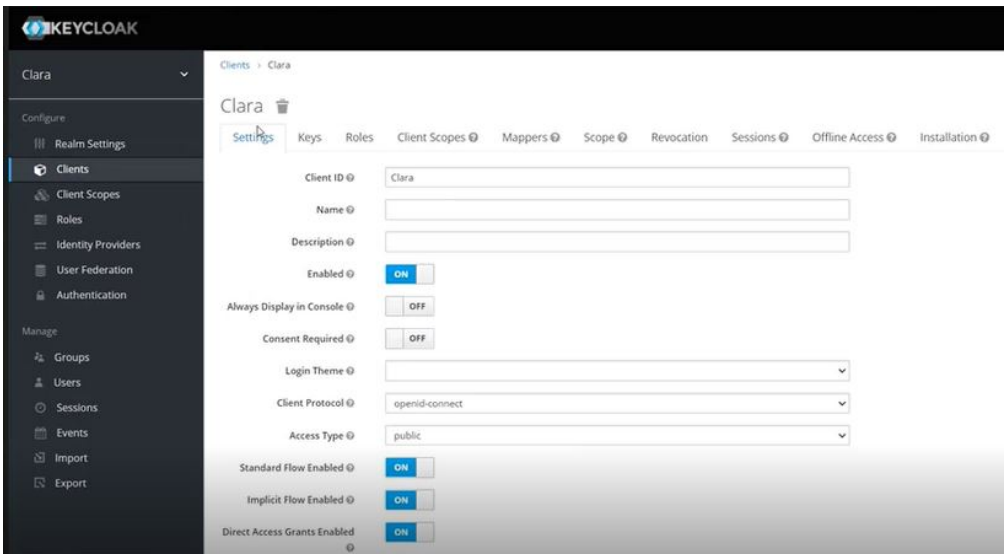
## Upgrade procedure

Run this procedure if you want to migrate your previously defined Keycloak settings and users:

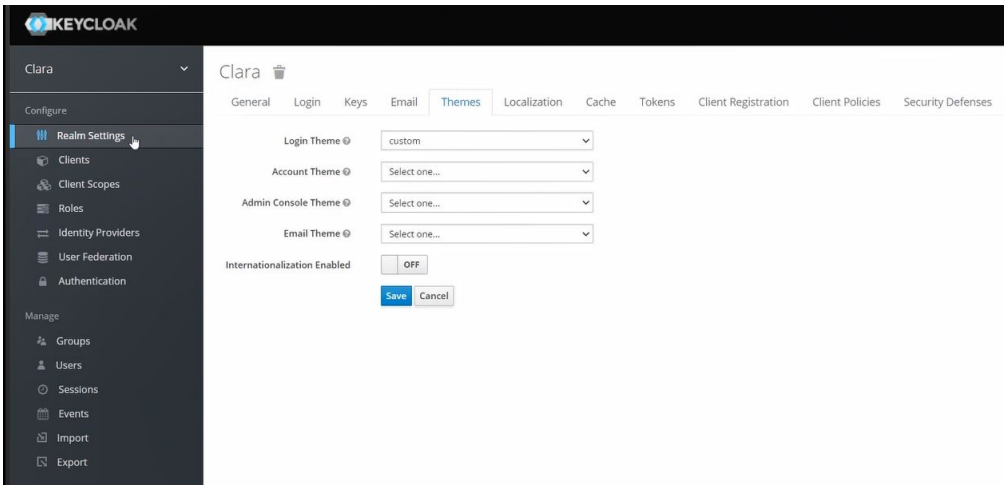
1. Install Clara V2.0.0.0 by following the installation procedure.
2. Open Keycloak administration console at the link **https://<IP:PORT>/keycloak/auth/admin** by using the following credentials:
    a. userid=**admin**
    b. password=**HclSolutions00**
3. Modify the login settings:
    a. From Keycloak left-hand sidebar, from the **Configure** section, select **Clients**
    b. Select **Clara** as **Client ID**



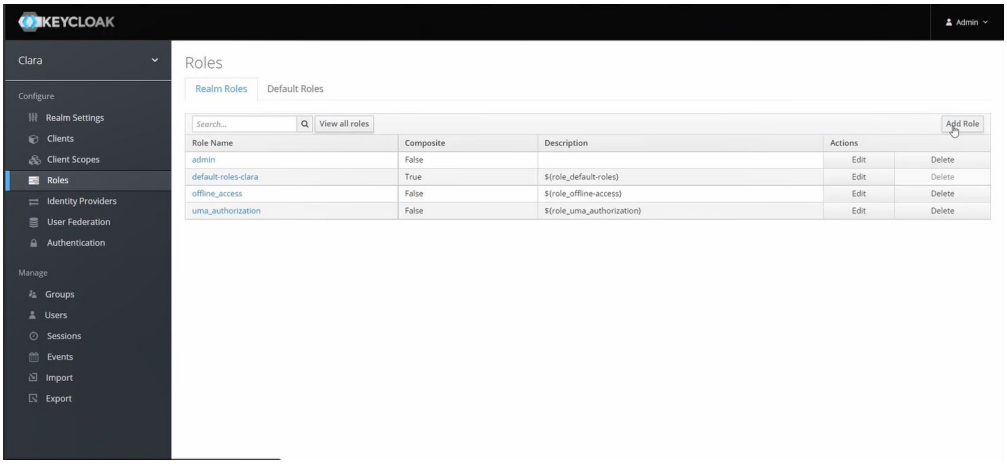

    c. From the **Settings** tab, set **Access Type** to *public*.



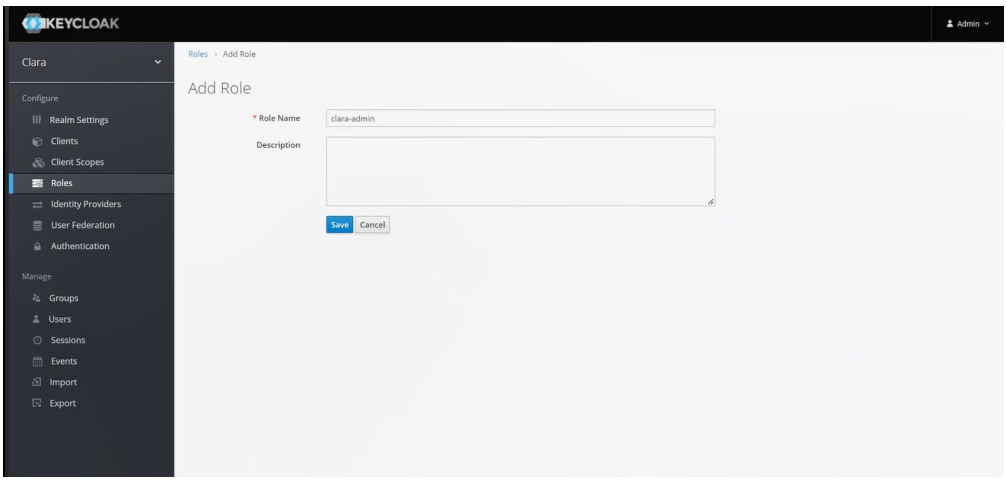

4. Modify the login theme:
    a. From Keycloak left-hand sidebar, from the **Configure** section, select **Realm Settings**
    b. From the **Themes** tab, set **Login Theme** to *custom*.




5. Add the new role: clara-admin:
    a. From Keycloak left-hand sidebar, from the **Configure** section, select **Roles**
    b. From the **Realm Roles** tab, click **Add Role**

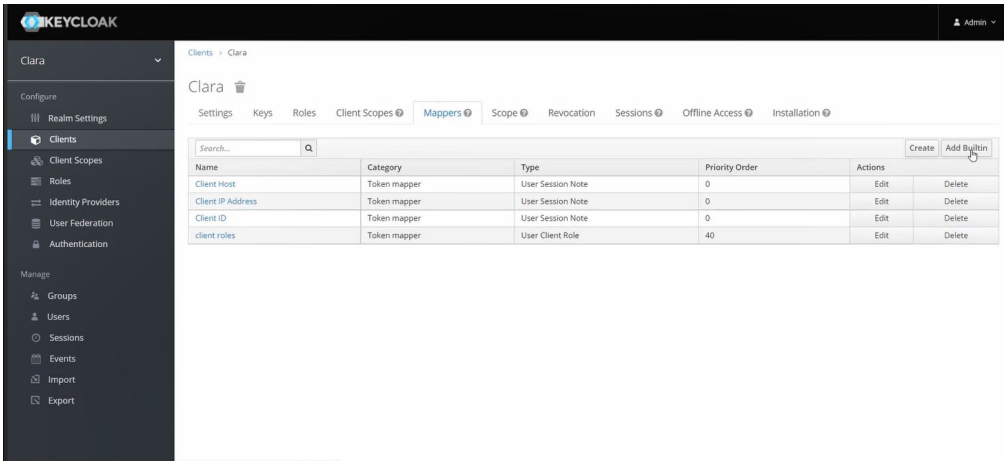



    c. Set **Role Name** to *clara-admin*

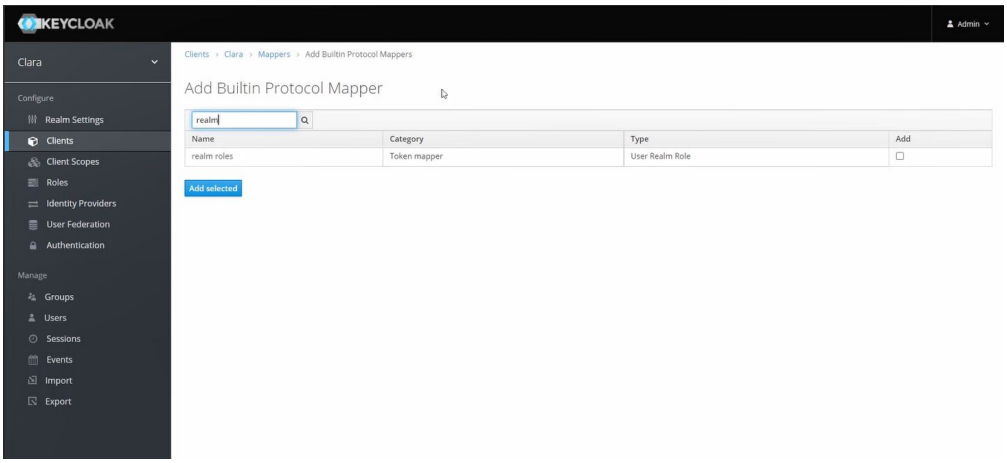



d. Click **Save**

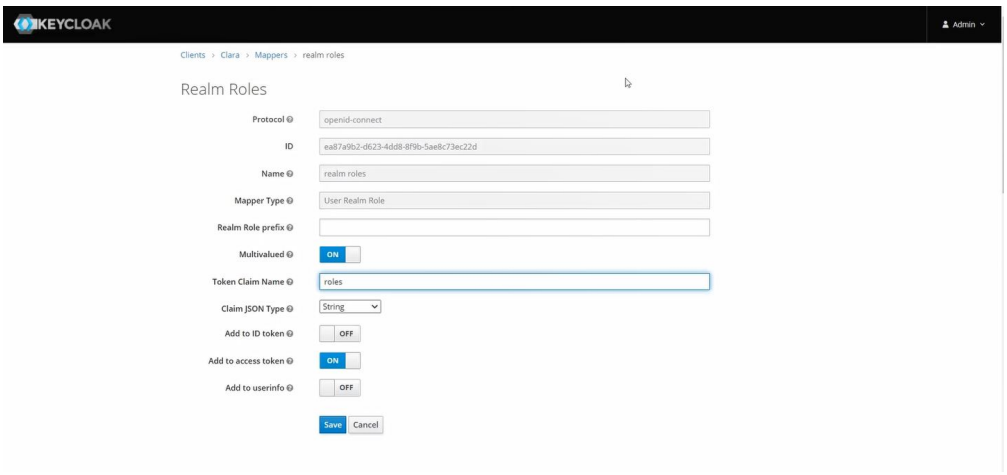
6. Modify the **Realm token**:
   a. From Keycloak left-hand sidebar, from the **Configure** section, select **Clients > Clara**
   b. From the **Mappers** tab, click **Add Builtin**



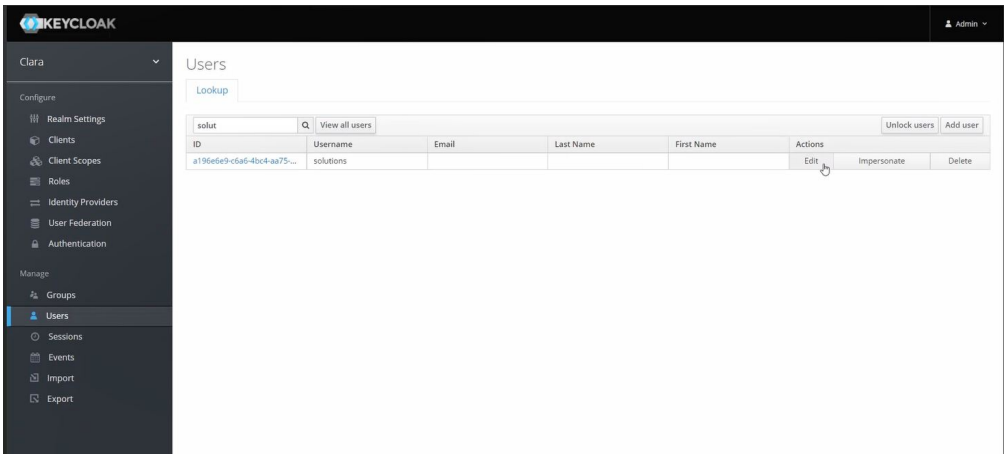

   c. Search for **realm roles** and click **Add selected**



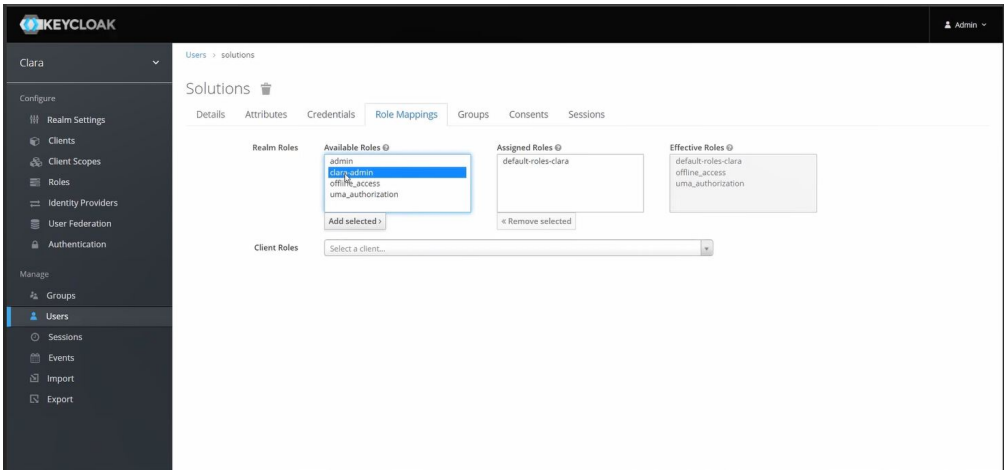

   d. From the **Mappers** tab, click the **Edit** action for **realm roles**
   e. Set the **Token Claim Name** to *roles*




7. Assign the new role *clara-admin*:
    a. From Keycloak left-hand sidebar, from the **Manage** section, select **Users**
    b. Search for the user you want to give the *clara-admin* role and click the **Edit** action




    c. From the **Role Mappings** tab, in the **Available Roles** list, select *clara-admin* and click **Add selected**




8. You can now login to Clara with your migrated users

After the upgrade, from **Clara Dashboard > Manage Intents,** you can import the .csv file containing the custom FAQs exported from your previous installation.

# 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 of 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**

   You can also reach the Keycloak  administration console from Clara Dashboard, by clicking **Manage roles** in the Account icon drop-down menu on the dashboard header.
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).
- 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**

# Accessing Clara Dashboard

Starting from V2, a Dashboard is available to simplify and speed up Clara configuration. Clara Dashboard is automatically installed together with Clara.
After Clara installation, you can access Clara Dashboard in the following ways:

1. At the link **https://<IP:PORT>/dashboard**
   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).
2. From the drop-down menu of Clara chat window.

Tasks you can run from Clara Dashboard are:

Managing the intents

Managing User Credentials

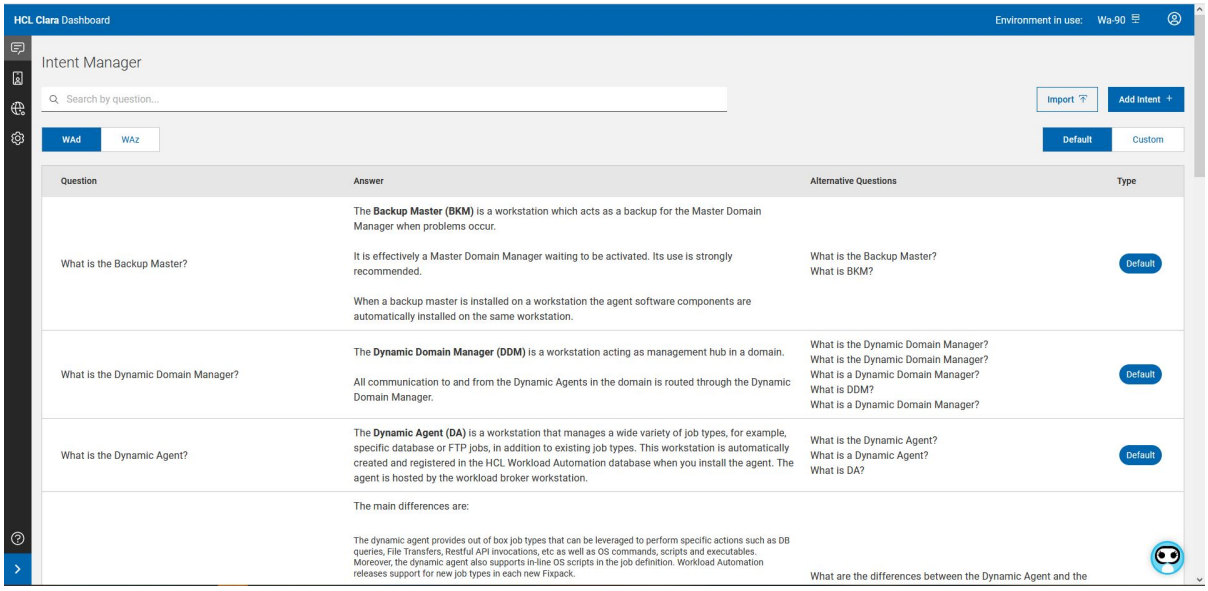
Managing the Workload Automation environments

Customizing Clara settings

# Managing the Intents

Through **Clara Dashboard > Intent Manager** page you, as an Administrator, can manage FAQs (or intents) for both Workload Automation and Workload Automation for Z, and add your own **custom** FAQs  to the **default** ones that you get with Clara.

In the world of virtual assistants, “intent” refers to the user's intention, that is his goal or requirement for engaging with a virtual assistant.




From the search bar, you can search for one or more intents.
Default intents cannot be edited or deleted.
Custom intents can be added, edited, deleted, and downloaded.
To add new intents, you can proceed in two different ways:

- Add one intent at a time:
  1. Click the **Add Intent** button.
  2. A side panel appears where you can enter the required information in markdown language.

- Add multiple intents at a time:
  1. Prepare a **.csv** file with three columns with title "description", "response", and "learn", containing the intents you want to add. Use the markdown language to format the text. Insert an intent for each row.

| description | response | learn |
|---|---|---|
| How can I get alert notifications from AIDA? | You can get alert notifications from AIDA in the following ways:<br><br>1. Through the **Anomaly Widget** on the Workload Dashboard<br><br>2. Via email<br><br>For details, see: [Receiving alert notifications](https://www.ibm.com/docs/en/workload-automation/10.1.0?topic=aida-receiving-alert-notifications) | How can I be notified by AIDA when an anomaly occurs?<br><br>How does AIDA notify me when anomalies occur?<br><br>How do I receive alert notifications from AIDA?<br><br>In which ways AIDA notifies me about alerts? |

2. Click the **Import** button to upload the file.

In both ways, for each intent you want to add, specify the following information:

| Field | Description | Note |
|---|---|---|
| description | The question you want to ask to Clara | |
| response | The answer Clara must provide | |
| learn | Insert different questions compatible with the provided answer | Needed for Clara training |

To export custom intents to a .csv file, select the intents and run the **Download** action.

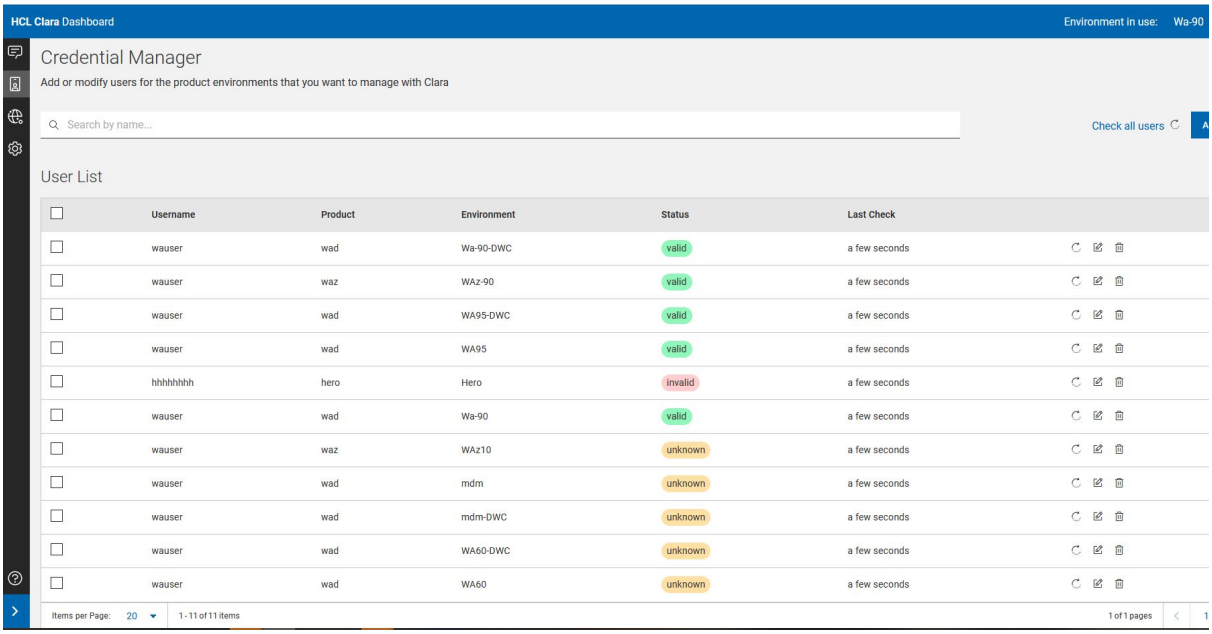
# Managing User Credentials

The **Credential Manager** component is accessible from **Clara Dashboard**. It helps you map your Clara credentials with the credentials of the following product environments:

- Workload Automation
- Workload Automation for Z
- Dynamic Workload Console
- HERO

As you start chatting with Clara, you must add the credentials of the product environments managed by Clara. Valid credentials are needed to invoke the REST APIs of the target product environment, thus enabling Clara to run actions on your behalf.

If you ask Clara to run actions on your product environments but your credentials have not been added yet, or are invalid, Clara alerts you and suggests to open the Credential Manager user interface to add or modify user credentials.




In the User Credential Manager main page, you can find the following information:

- **Username** - the username you use to access Clara.
- **Product** - the product you want to manage with Clara. It can be: WAd | WAz | DWC | HERO.
- **Environment** - The name of the product environment (for example, PROD1) as defined in **Clara dashboard > Settings > Environments**.

- **Status** - 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.
- **Last check** - The number of days since the last update of the user credentials for your product environment.

Actions you can run on each user are:

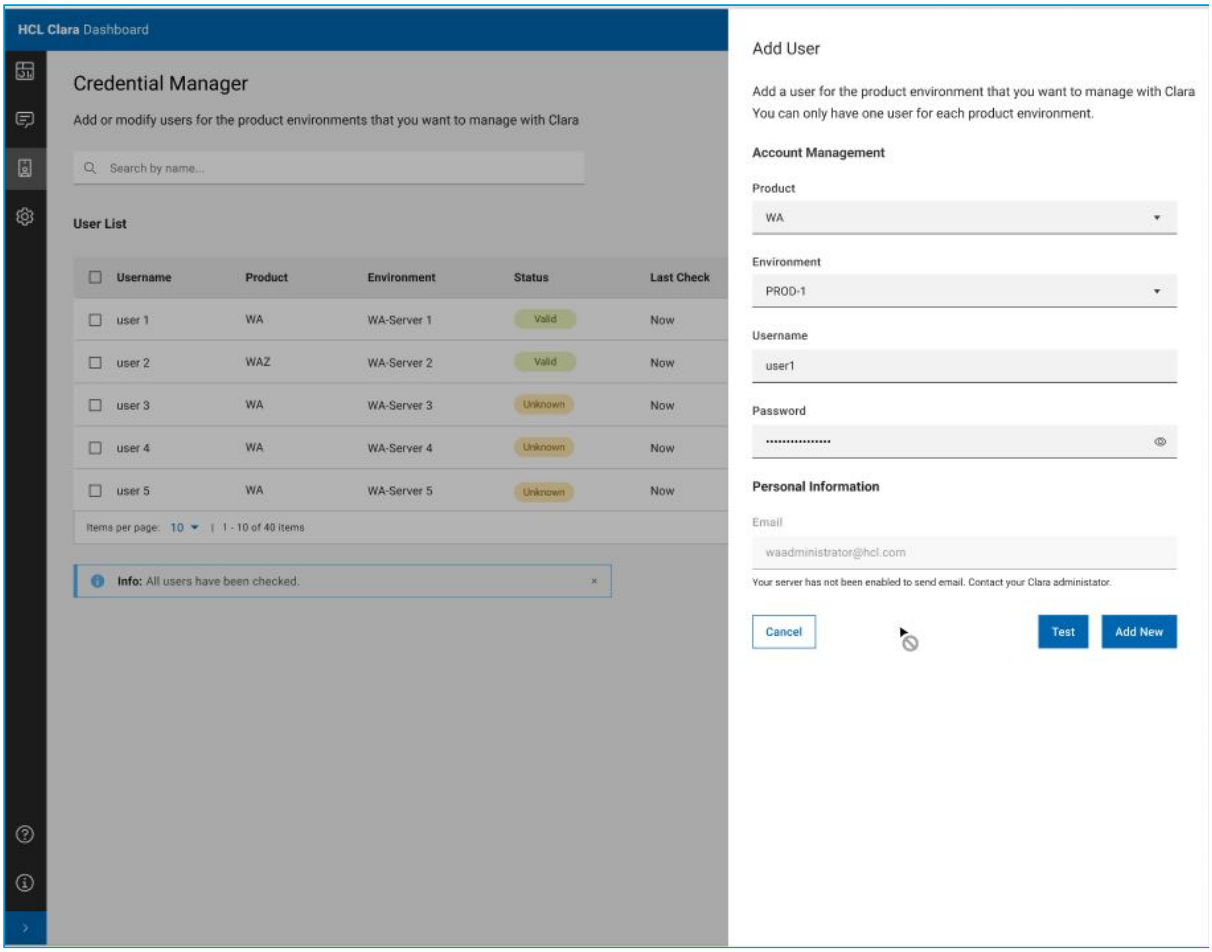
- **Edit user** - to modify a user.
- **Check credentials** - to check if the user credentials are valid.
- **Delete user** - to delete the user credentials.

If the status of some users is **Unknown,** click the button **Check all users** to validate all user credentials and update their status.
To add a user, click the button **Add new**. For details, see Adding a user.

## Adding a User

You can add a user for the product environments that you want to manage with Clara. Each Clara user can only add a user, with his set of credentials, 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.**
2. The Add User side panel opens:




3. Select a value for Product and Environment.
4. Provide Username and Password for the environment.
5. Provide your e-mail address to receive notification e-mails from Clara.
6. Click **Test** to check user credentials.
7. If user credentials are valid**,** click **Add new**.
8. If it is not possible to check the credentials because, for example, the connection with the target environment is down, you can save the new user with an unknown status for credentials and verify them later on.
9. You can save the user even if the 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:
2. Verify if the status of your user credentials.
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 side panel opens:
3. Modify Username, Password, and e-mail information.
4. Click **Test** to check the user credentials.
5. If the user credentials are valid, click **Add new**.
6. You can save the user even if credentials are invalid or their status is unknown, and modify the credentials or check their status later on.

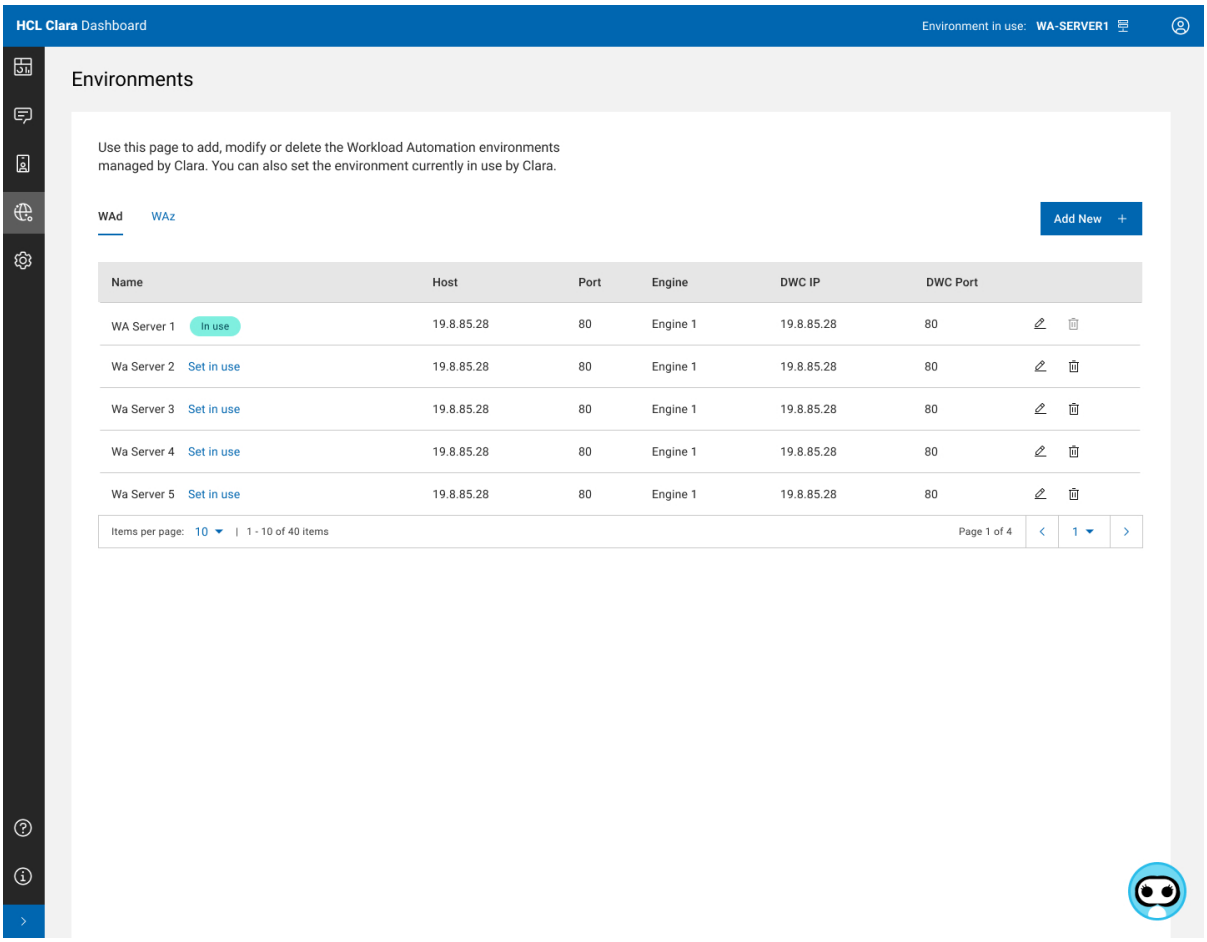
## Deleting a user

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

# Managing the Workload Automation environments

Through **Clara Dashboard > Environments** page you can add, modify, or delete the Workload Automation environments managed by Clara. You can also set the environment currently in use by Clara.




Two environment tables are available and you can switch from one to another:

- **WAd table**, showing the environments available for HCL Workload Automation (WAd).
- **WAz table**, showing the environments available for HCL Workload Automation for Z (WAz).

The environment that is **currently in use** by Clara is highlighted in turquoise.
Actions that you can run on each environment are:

- **Set in use** - to switch from the currently used environment to another.
- **Edit** - to edit the environment configuration parameters.
- **Delete** - to delete the environment. You cannot delete the environment currently in use.

Click the **Add New** button to add a new environment.
Click the **Test all** button to check the connectivity between Clara and the available environments.

For each **WAd** environment that you want to add, specify the following properties:

| Property | Description | Notes |
|---|---|---|
| Type | The environment type. It must be "WAd". | |
| Name | The name of the environment. | |
| IP Address | The IP address of the Workload Automation engine. | |
| Port | The port to connect the Workload Automation engine. | Default: 587. |
| Engine Name | The name of the engine definition within the Workload Automation environment. | |
| DWC IP Address | The IP Address of the Dynamic Workload Console. | Optional, if you want to define a DWC within the environment. |
| DWC Port | The Port of the Dynamic Workload Console. | Optional, if you want to define a DWC within the environment. |

For each **WAz** environment that you want to add, specify the following properties:

| Property | Description | Notes |
|---|---|---|
| Type | The environment type. It must be "WAz". | |
| Name | The name of the environment. | |
| DWC IP Address | The IP Address of the Dynamic Workload Console. | |
| DWC Port | The Port of the Dynamic Workload Console. | Default: 587. |
| Engine Name | The name of the WAz engine to which the DWC is linked. | |

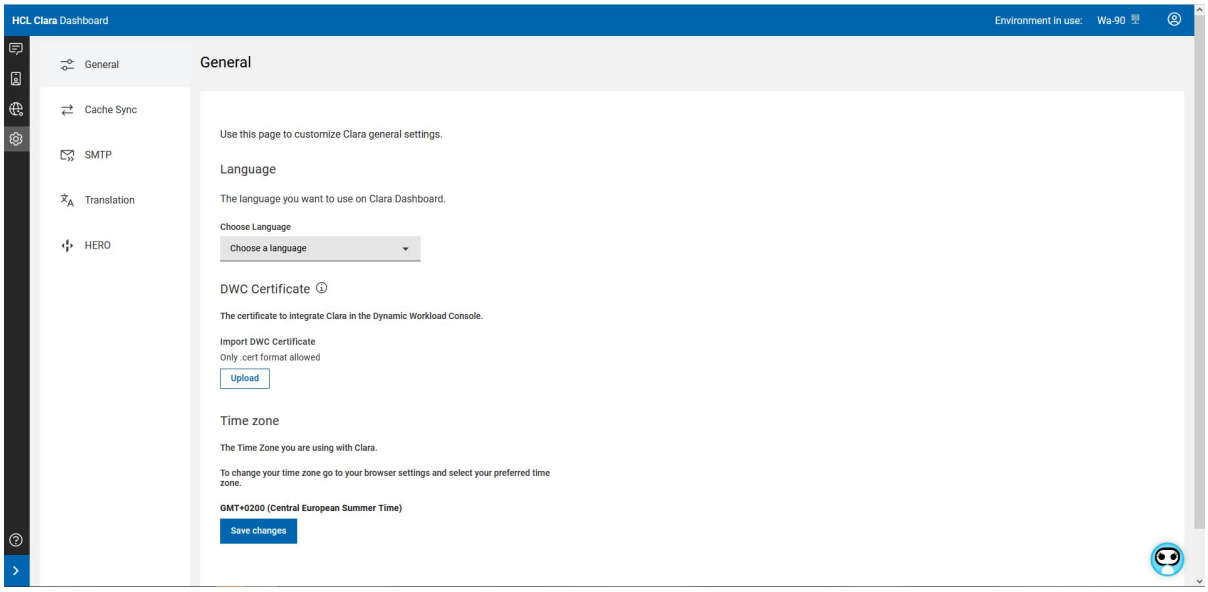
The credentials to access each Workload Automation environment and the related Dynamic Workload Console must be defined in the **Dashboard > Credential Manager.**

# Customizing Clara settings

After Clara installation, from **Clara Dashboard > Settings** page you can customize the following settings:

- General
- Cache synchronization
- SMTP
- Translation
- HERO integration

Select the corresponding icon on the Settings left hand panel.




## General

Use this page to customize the following Clara general settings.

| Property | Description | Notes |
|---|---|---|
| Language | The language you want to use on Clara Dashboard. English, Spanish, German, Portuguese, and Italian are supported. | This language might be different from the language you use on Clara UI. |
| DWC Certificate | The certificate to integrate Clara in the Dynamic Workload Console. Click **Upload** to import a .cert file containing the certificate. | In Clara, this certificate is set to the default certificate of the Dynamic Workload Console. If a custom certificate is used instead, |

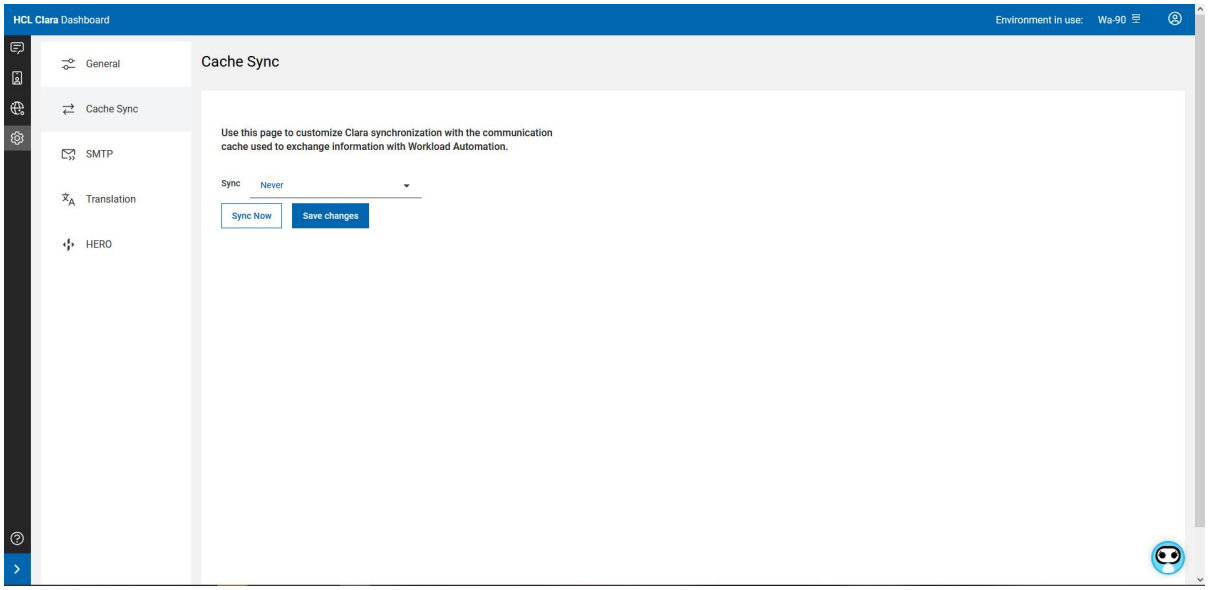
| | | |
|---|---|---|
| | | you must modify this parameter accordingly. |
| Time Zone | The Time Zone you are using with Clara. | To change the time zone go to your browser settings and select your preferred time zone. |

**Note:**
For the steps to run on the Dynamic Workload Console, see the following topics in the **Workload Automation** documentation:

- *Integrating Clara in the Dynamic Workload Console*
- *Integrating Clara after an environment update*

## Cache Synchronization

To reduce the response time, Clara V2 uses a dedicated cache to exchange information with Workload Automation. Use this page to customize Clara synchronization with the communication cache.



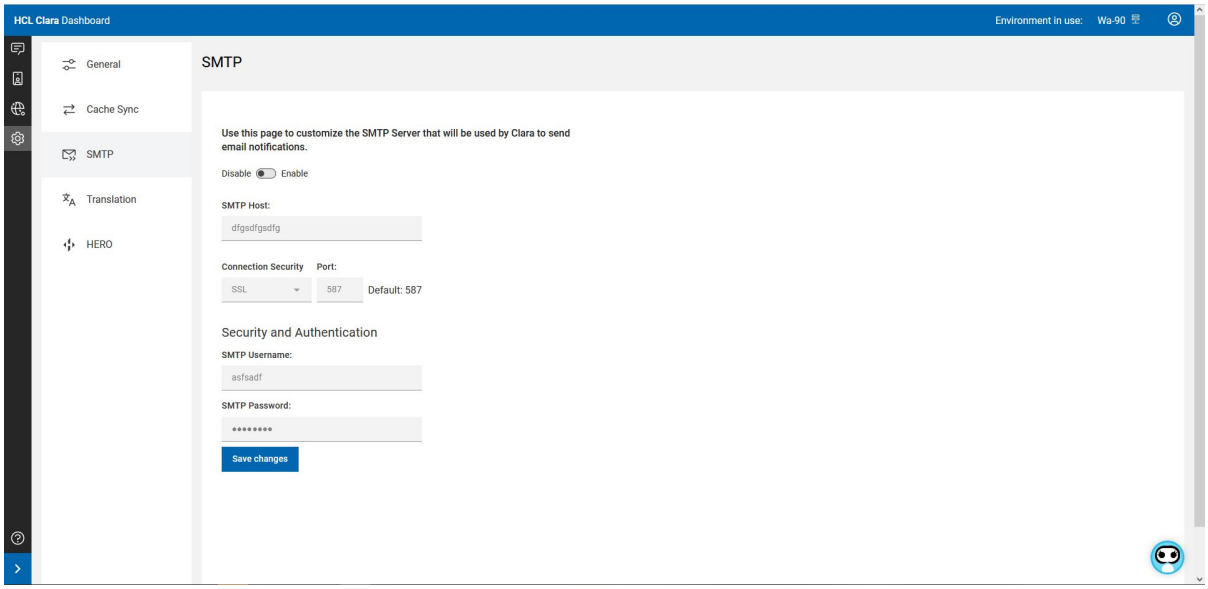

You can specify how often you want to synchronize Clara with the communication cache. Select one of the following values:

- Never
- Now
- Hourly, every x hours
- Every day at *hh:mm*
- Every week, on *day,* at *hh:mm*

While a synchronization is in progress, you can stop it. Any progress will be lost. You can restart the synchronization later on.

## SMTP

Use this page to customize the SMTP Server that will be used by Clara to send email notifications.

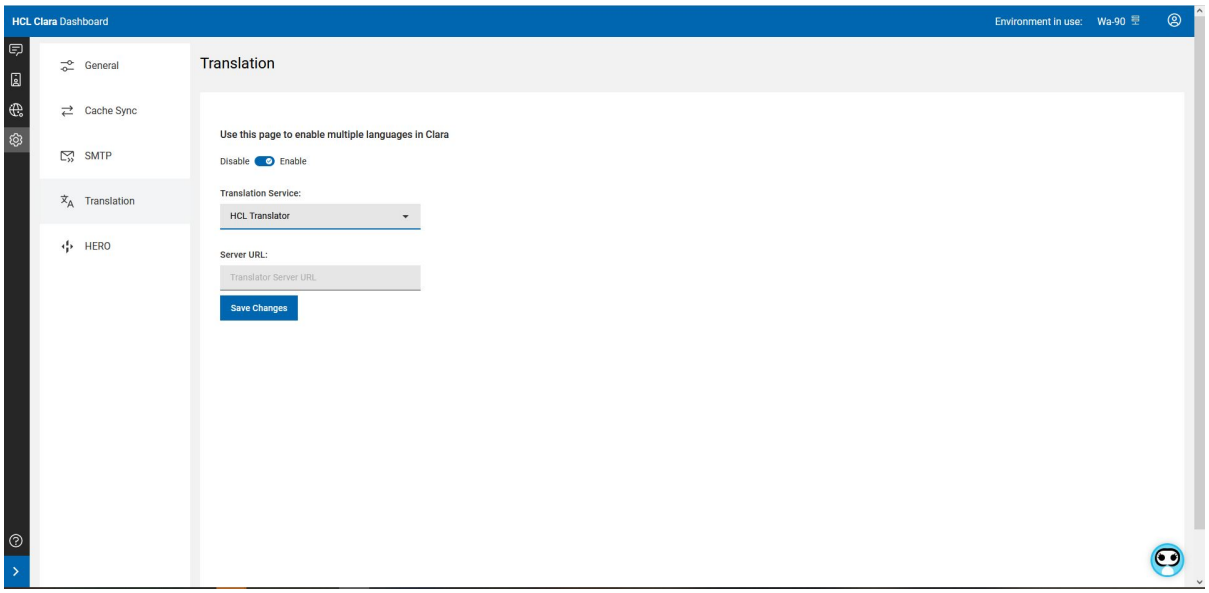



You can enable/disable this functionality. If you choose to enable it, you must specify the following properties:

| Property | Description | Notes |
|---|---|---|
| SMTP Host | The fully qualified hostname of the SMTP Server. | Example: smtp.gmail.com. |
| Connection Security | The authentication type defined for the SMTP Server. Can be TLS or SSL. | Default:TLS. |
| Port | The Port of the SMTP Server. | Default: 997. |
| SMTP Username | The username sending email to the recipient specified in the Clara Credential Manager. | Example: username@gmail.com. |
| SMTP Password | The password associated to the username. | |

## Translation

Use this page to enable or disable the availability of multiple languages in Clara.




Clara V2 provides **multilanguage enablement** through any the following Translation Services:

- HCL Translator (default)
- Google
- Azure

**Note**: Multilanguage enablement in Clara is obtained through machine translation. Therefore, it could sometimes produce inaccurate results.
If you are using HCL Translator and you find that the translation is not accurate, you can submit your feedback through the **HCL Software portal** by providing the original sentence and the correct translation.
In this way, the HCL Translator development team can use your input to re-train the language model and improve the translation quality.

**HCL Translator** is an API based service. It is based on Machine Learning models that have been trained and fine-tuned to understand context-specific terms or phrases used in the Workload Automation domain, thus greatly improving the user experience. It supports bidirectional automatic translation from English to French, German, Italian, Portuguese, and Spanish. For details, see: Installing and Configuring HCL Translator.

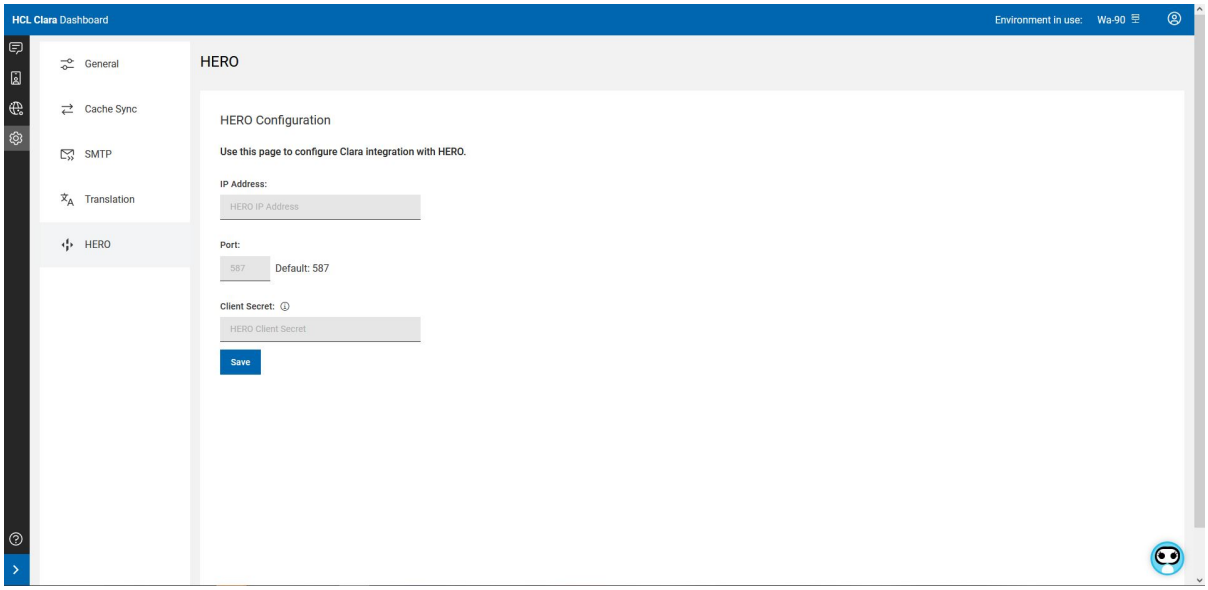
After Clara installation, you can enable/disable the support of multiple languages. If you choose to enable them, you must specify the following properties:

| Property | Description | Notes |
|---|---|---|
| Translation Service | Can be: "Google", "Azure", or "HCL Translator". | Default: HCL Translator. |
| **For Google:** | | |
| Server URL | The URL of the translation service provided by Google. | |
| Languages | The languages you want to use with Clara. | |
| Authentication file | The file that stores the Google translation service account key. | Must be a json file. For details, see: https://cloud.google.com/translate/docs/setup. |
| **For Azure:** | | |
| Server URL | The URL of the translation service. | |
| Subscription Key | The subscription key of the Azure translation service. | For details, see: https://docs.microsoft.com/en-us/azure/cognitive-services/translator/quickstart-translator. |
| Region | The region where the Azure Server is located. | Default: "switzerland nord". |
| Languages | The languages you want to use with Clara. | |
| **For HCL Translator:** | | |
| Server URL | The URL of HCL Translator provided by SoFy after the HCL Translator installation. | |

## HERO integration

Only for WAd type environments, use this page to configure Clara integration with HERO.

Clara can monitor and get information about your Workload Automation servers through HERO (HEalthcheck & Runbook Optimizer). Clara can monitor one environment at a time. For details, see: Clara monitors your servers through HERO.




Specify the following properties for the environment that you want to monitor:

| Property | Description | Notes |
|---|---|---|
| IP Address | HERO server IP address. | |
| Port | HERO server Port. | Default: 587. |
| Client Secret | The HERO Client Secret to open a HERO session from Clara. | For details about HERO Client Secret, see the topic: "Configuring security" in HERO documentation. |

The credentials to access HERO must be defined in the **Dashboard > Credential Manager.**

Click the **Add New** button and select **HERO** in the **Product** drop-down list.

# 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. You must enable notifications in your browser (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 have problems in receiving notification on Mac:

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

# About the User's Guide

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

Interacting with Clara

Clara replies to your questions

Clara runs actions on your behalf

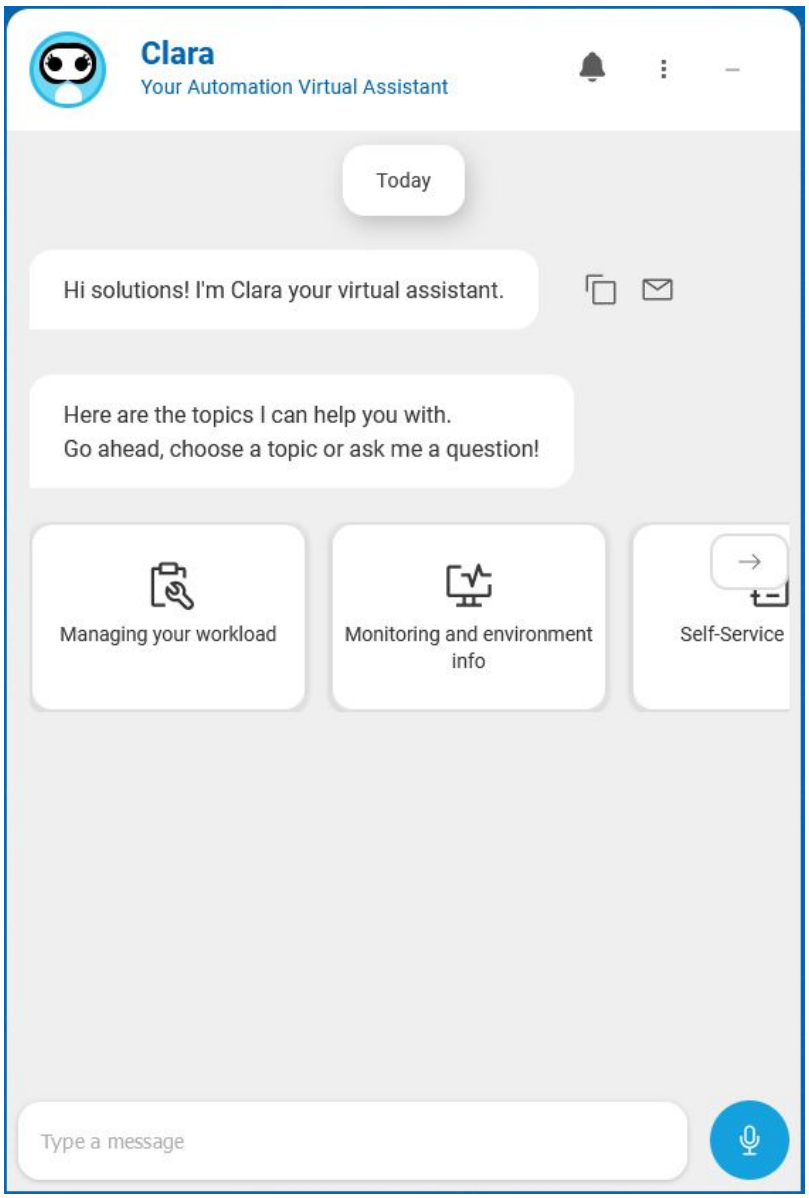
Clara monitors your servers through HERO

Clara helps in problem determination

# Interacting with Clara

You can chat with Clara through the following user interfaces:

- Clara web interface, available at the following link **https://<IP:PORT>** 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).
- Clara integrated in the Dynamic Workload Console of Workload Automation. For details about configuration instructions, see Installing and configuring Clara.
- **Slack** and **Teams** messaging tools. For details, see Installing and configuring Clara.



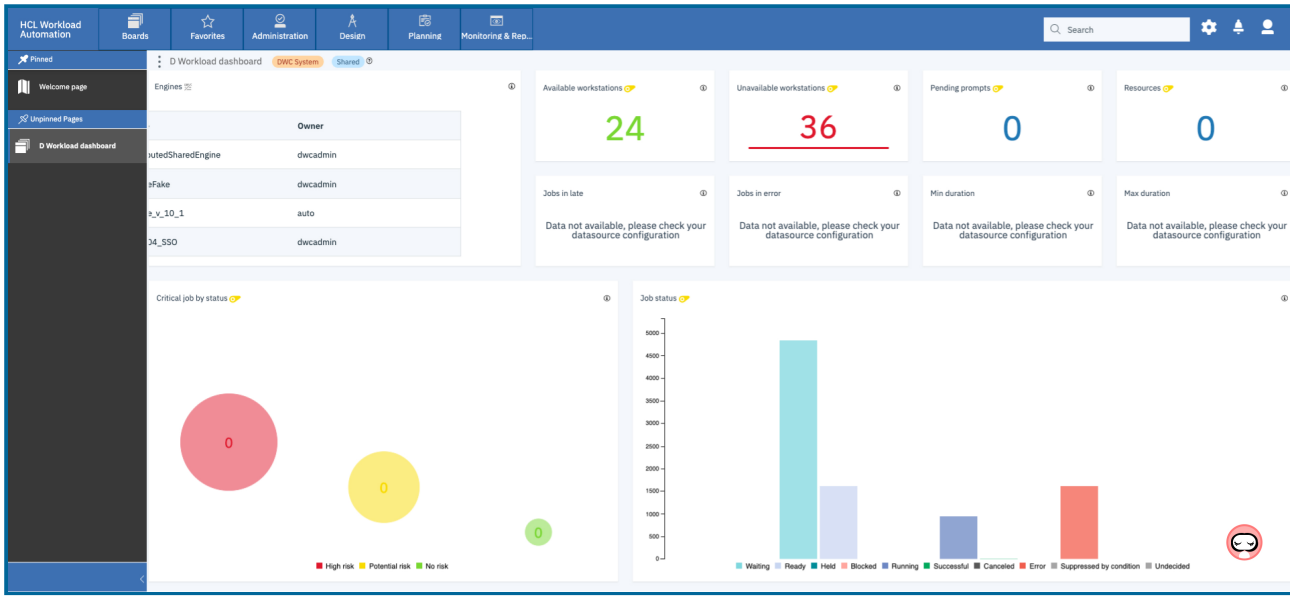
## Clara avatar

Clara avatar has been redesigned to show emotions, so users can immediately understand the status of an action.


Clara can switch from a status to another of the following:

- **Idle status** is used when Clara is in neutral status or there are no errors to show.
- **Success status** is used for actions that  complete successfully (for example, "Monitoring a job").
- **Error status** is used for actions that  cannot complete or complete with errors.

When integrating Clara in the Dynamic Workload Console, even when Clara's chat window is closed, the avatar can show you if an action completes successfully or anything goes wrong in the background.



## Selecting a conversation language

After the installation, to enable the availability of multiple languages with Clara, you must configure the translation service of your choice (see Translations). The first time you chat with Clara, she asks you to select a conversation language from the set of languages defined during the configuration steps.

You can change the conversation language at any time, by selecting **Change Language** in the drop-down menu of Clara chat window.

If you are using a language other than English, you can view the English version of a message by clicking **Show in English** in the message box.

## How to start a conversation with Clara

To start a conversation with Clara on a specific topic, proceed as follows:

- When you login to Clara, select one of the topics suggested by Clara. For an overview of the topics, see Running actions on your behalf.
- When the conversation flow about a topic completes, type "*What can you do?***"** or click the **What can you do?** button, to view other topics Clara can assist you with.
- Simply ask your free question (for example, *What is a job?* or *Current plan status*). For details, see Clara replies to your questions.
- Ask Clara to help you solve a problem (for example, *Why a job does not start?*). For details, see Clara helps in problem determination.

Clara maintains a history of the messages you have entered. By using the up and down arrow keys, you can recall previously-entered messages to the input field.

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

- **Export conversation** - to export the current conversation with Clara into a .txt file.
- **About** - to get information about Clara version, release, and terms of license agreement.
- **Open dashboard** - to open Clara Dashboard for configuration.
- **Change language -** to select a different conversation language.
- **Open documentation** - to open Clara documentation.
- **Clear conversation** - to reset the current conversation with Clara.
- **Logout** - to log out from Clara.

During a conversation, Clara provides **tips,** in the form of action buttons, to suggest the next action you can run.
To quit a conversation, type **Cancel**.

## Copy and share Clara's messages via email

When chatting with Clara, by hovering over Clara's messages, two options are available for each message: **Copy** and **Send email**
By selecting **Copy**, you can copy the message to clipboard, and paste it later on.
By selecting **Send 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.
Clara answers "*What can I do?"* and "*How can I do it?"* types of questions by leveraging a specialized Knowledge Base, and also provides useful links to videos, blogs, articles, and product documentation.
The current Knowledge Base includes answers to the most Frequently Asked Questions(FAQs) about Workload Automation, many of which come from Customer Support cases and are certified by Workload Automation specialists.
See below for examples about frequently asked questions handled by Clara.

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:

- Workload Automation documentation
- Automation Hub documentation

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 runs actions on your behalf

At Clara startup, or when you enter the "*What can you do?***"** conversation flow, you can view the **smart cards** representing the categories of actions that Clara can run on your behalf.

1. **Managing your workload** - You (as an operator) can submit a job or modify a job execution by running the following actions:
    - **Rerun a job** - to rerun a job that is in error status (available for WAd only).
    - **Submit a job -** to submit a predefined job (available for WAd only). For details, see Submit a job.
    - **Submit a job stream** - to submit a predefined job stream for processing (available for WAd and WAz).
    - **Cancel a job** - to cancel a job before it is launched (available for WAd only).
    - **Release a job** - to release a job from dependencies (available for WAd and WAz).
    - **Hold a job** - to put a job in HELD status, awaiting for dependencies resolution (available for WAd and WAz).
    - **Change the priority of a job** - to change the order of running of a job (available for WAd only).
    - **Release all dependencies of a job** - (available for WAd only).
    - **Change the limit of a workstation** - to change the maximum number of jobs running concurrently on that workstation (available for WAd only).
    - **Link a workstation** - (available for WAd only).

2. **Monitoring and environment info** - You (as an administrator) can get information about scheduled jobs by running the following actions:
    - **Get the log of a job** - to view the log of a job and download it.
    - **Get the status of a job** - to retrieve information about the current status of a job. For the completed jobs, you can get the job log or rerun them if completed with errors (available for WAd and WAz).
    - **Monitor a job** - to monitor a job and receive a notification when the job status changes (available for WAd only). For details, see Monitor a job.
    - **Get the overall status of the current plan** - to retrieve information about the status of all the jobs in the current plan (available for WAd and WAz). For details, see Get the status of the current plan.
    - **Get the status of a workstation** - to search for a workstation and see its status.

- **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 the workload in the Self-Service Catalog by running the following action:
   - **Submit a service from the catalog** - (available for WAd and WAz). For details, see Submit a service from Self-service Catalog.

4. **Troubleshooting** - You (as an operator or administrator) can speed up the Workload Automation troubleshooting. For details, see Clara helps in problem determination. Actions you can run are:
   - **Why a job is not starting** - You can understand why a job is not starting (available for WA only).
   - **Get the jobs in error in the current plan** - You can find any jobs in error in the daily plan (available for WAd and WAz).
   - **Get the critical jobs in the current plan** - You can find any jobs in late in the daily plan (available for WAd and WAz).

5. **Monitoring with HERO -** You (as an administrator) can monitor and get information about your Workload Automation servers through HERO. For details, see Clara monitors your server through HERO. Actions you can run are:
   - **Get the overall status of PROD servers** - to monitor and get information about your WA production servers through HERO (available for WA only).
   - **Are there any issues with my servers?** - to quickly discover if there are any issues (errors or warnings) with your production and non production servers.
   - **Monitor a server** - you can ask Clara to monitor a specific server and send you a notification when the server status changes.

**Note:** For details about how to search for a job when running one of the above action, see **Searching for a job.**

When you run an action, Clara leverages the authentication mechanism of Workload Automation. If a user is not authorized to run an action in Workload Automation, Clara returns an error.

## Searching for a job

For all the actions dealing with jobs, such as *Submit a job* or *Monitor a job*, when you start the action, the conversation flow you have with Clara to identify a job is as follows:

1. Clara asks for the job name and shows the list of all jobs in the current Workload Automation environment.
2. Select a job from the list or, if you remember the job name (full or partial), type it in the input field. You can filter the job names in different ways:
    - By using the "*" wildcard anywhere in your search string. You can use more than one "*" wildcard in your search string.
    - By typing the name (full or partial) of the workstation where the job has been defined.
    - By typing the name of the job stream (full or partial) that contains the job.
    - By typing the name of the folder. For example, type */folder_name/** to get all jobs in the *folder_name* folder.

    Filters can be combined all together.

    Otherwise, click one of the following tips:
    - **Cancel** - to cancel the action
    - **Search by name** - to search the job by providing the name (full or partial) of:
        - The job itself.
        - The workstation where the job has been defined.
        - The job stream that contains the job.
        - The folder.
    - **Search by folder** - to search the job by folder name. If you know the folder name, type */folder_name/** in the input field to get all jobs in the *folder_name* folder.
    - **Show me more jobs** button to see more jobs.

The process of searching for an object is technically called **entity recognition process**. In Clara, it is based on data extraction and caching of the information related to your Workload Automation environment. In this way, Clara is always aware of the conversation context and can easily identify the names of jobs, job streams and workstations at any moment, making the conversation faster and smarter.

## Submit a job

This action is available for WAd type environments only.

To submit a job with Clara, run the following steps:

1. In the "*What can you do?"* conversation flow, select **Managing your workload > Submit a job.**
2. Identify the job to submit by following the Searching for a job conversation flow.
3. Clara asks for the job alias. Click the tip **No alias** if you don't want to set a job alias.
4. Clara asks for the job stream and shows a list of job streams for submitting the job.
5. Type a job stream name or select one from the list. You can filter the job stream names by using the "*" wildcard anywhere in your search string.
6. Clara asks for the workstation name and shows the list of the workstations in the current Workload Automation environment.
7. Type a workstation name or select one from the list. You can filter the workstation names by using the "*" wildcard anywhere in your search string.
8. Clara asks for a confirmation about the action on the selected job.
9. Click Yes to confirm.

Some steps of the above conversation might be skipped by Clara if, based on the entity recognition process, she is already aware of the job stream or workstation name.

At each step of the conversation flow, Clara provides the tip **Search by name.** By selecting this tip, Clara allows you to search for a job stream or a workstation by providing the full or partial name. You can filter the search result by using the "*" wildcard anywhere in your search string.

For an even quicker interaction, you can type your entire command in the input field. For example:
**Submit job** *job_name* **in** *job_stream_name* **on** *workstation_name* **with alia**s *alias.*

## Monitor a job

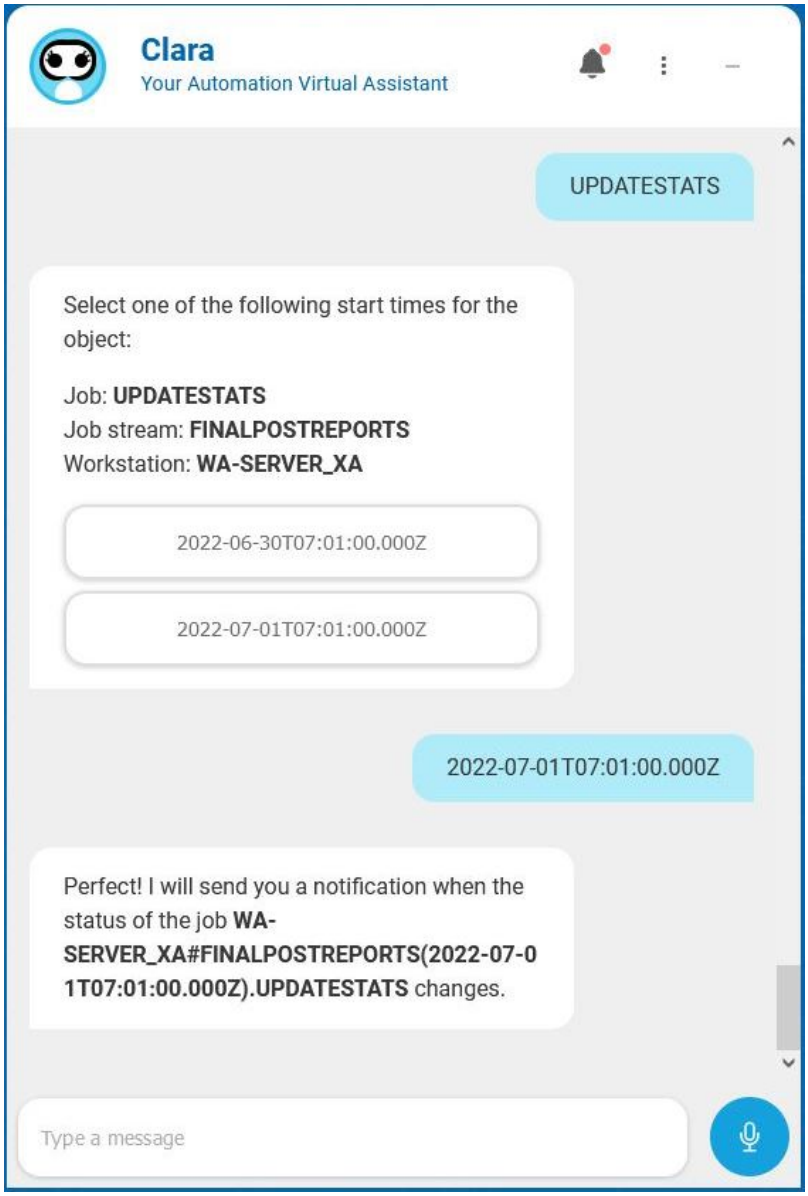
Clara can monitor a job and send you **notifications** when the job status changes.

To receive Clara notifications, you must first enable notifications in your browser. For details, see: How to enable persistent notifications. If you skip this step, Clara warns you with a message and provides a button to enable the notifications.

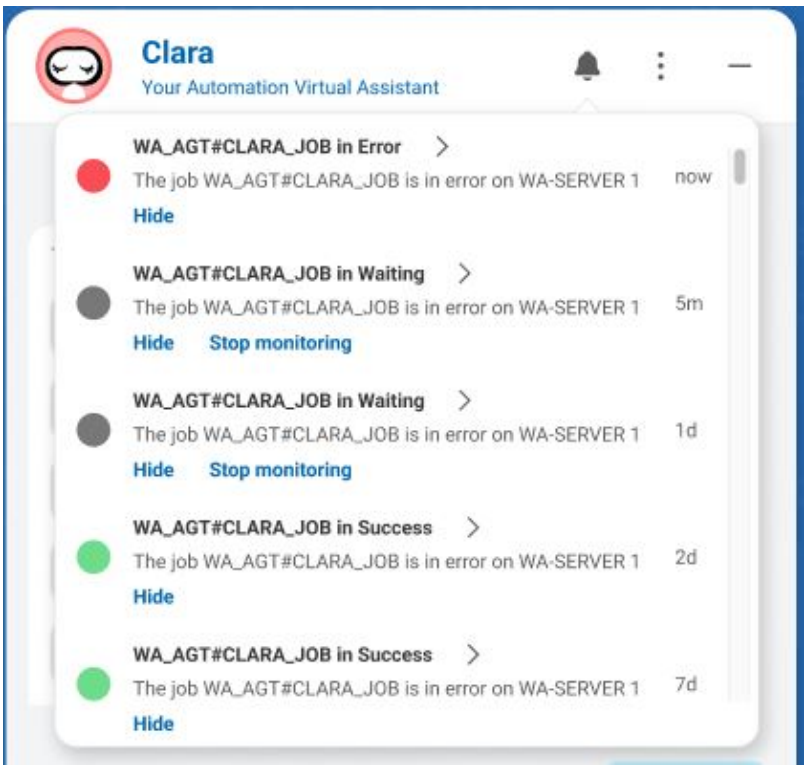
To monitor a job and receive Clara notifications, run the following steps:

1. In the "*What can you do?"* conversation flow, select **Monitoring and environment info > Monitor a job.**

2. When required by Clara, provide the job name, the job stream name, and the workstation name for the job you want to monitor. For details, see Searching for a job.
3. Clara starts monitoring the job. She checks the job status at time intervals depending on the estimated duration of the job, and notifies every status change.
4. A red dot appears on the bell on the chat header meaning that you have unread notifications. An e-mail with the new job status is also sent to the e-mail address you have specified during the **Add User** flow, on the Credential Manager user interface.




5. To view the notification messages, click the bell. A panel appears, showing the notification queue. A maximum number of 10 notifications is kept for a week.

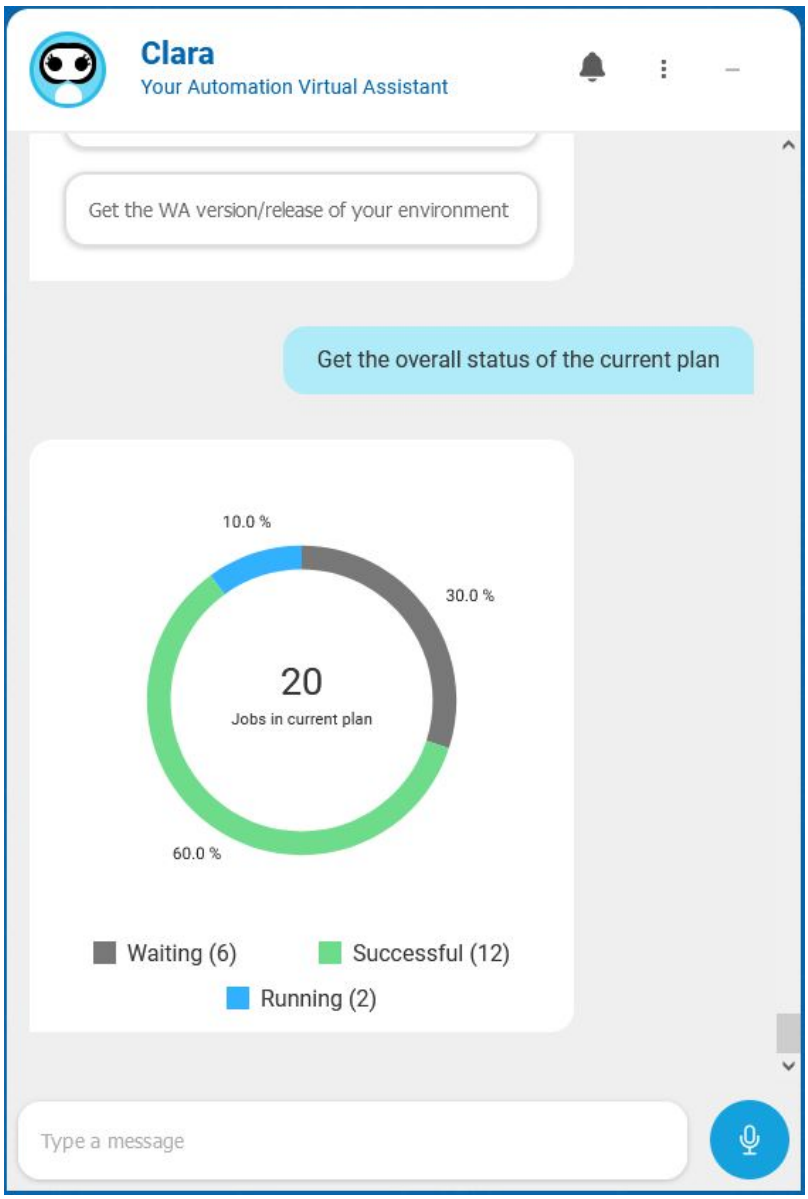



6. For each notification, you can run the following actions:
    - **Hide** - to hide the notification.
    - **Stop monitoring** (only for jobs not yet completed) - to stop monitoring the job.
7. By clicking on each notification, you can get the timestamp of when the status change occurred.

## Get the status of the current plan

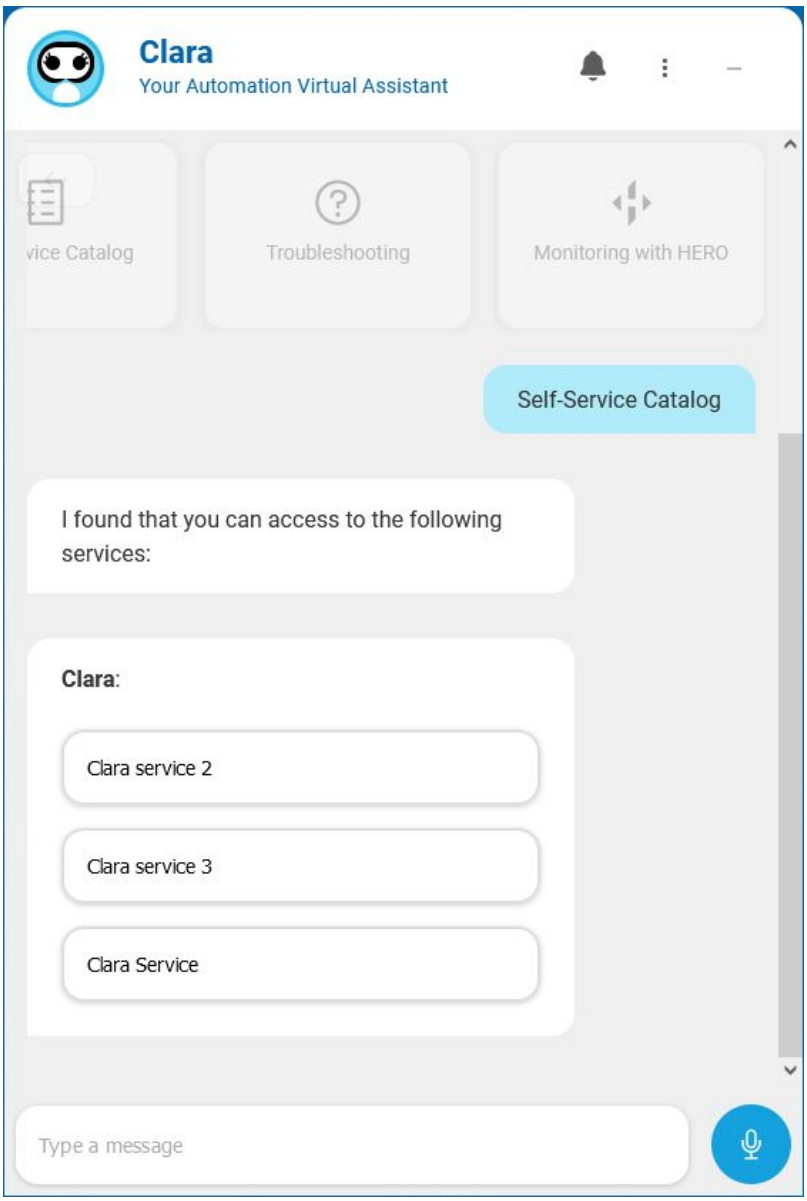
As a Workload Automation administrator, you might need to check the status of the current plan. Run the following steps:

1. In the "*What can you do?*" conversation flow, select **Monitoring and environment info > Get the overall status of the current plan**
2. Clara provides an interactive segmented donut chart where each color corresponds to a job status. By clicking each segment of the chart, you get the number of jobs in that specific status.

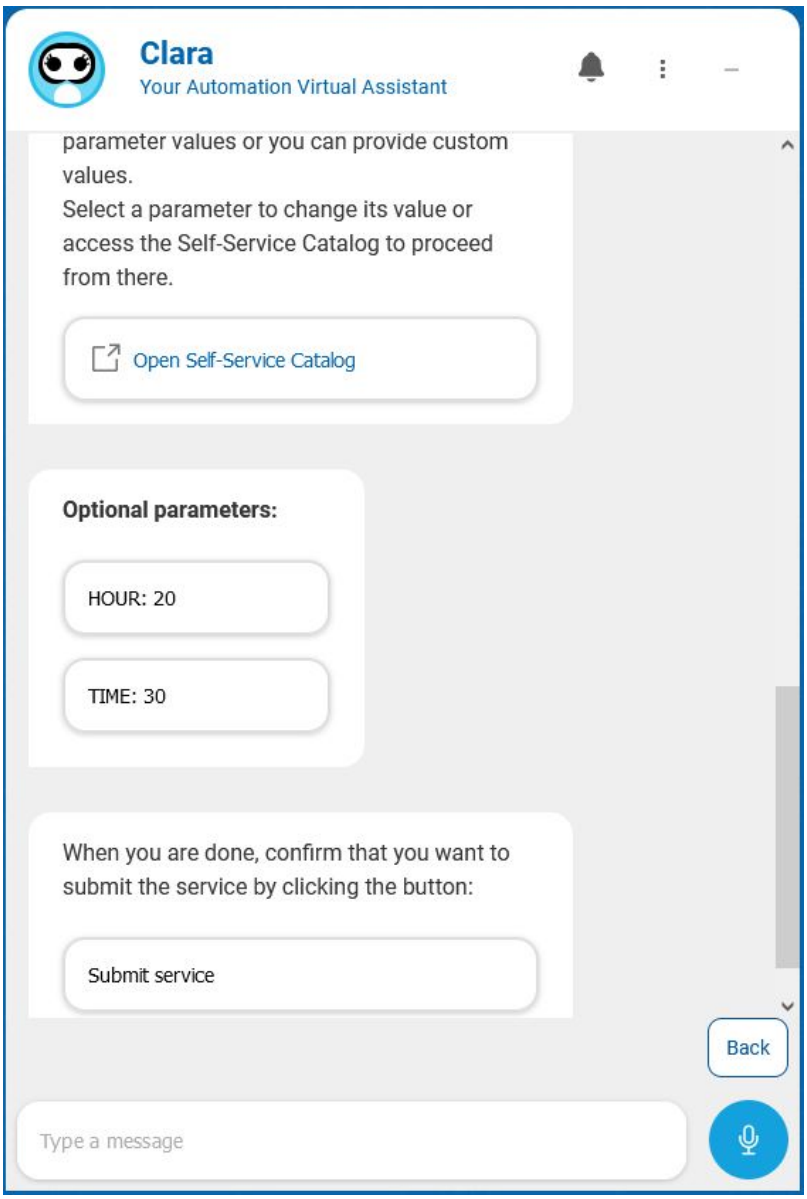



## Submit a service from Self-service Catalog

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

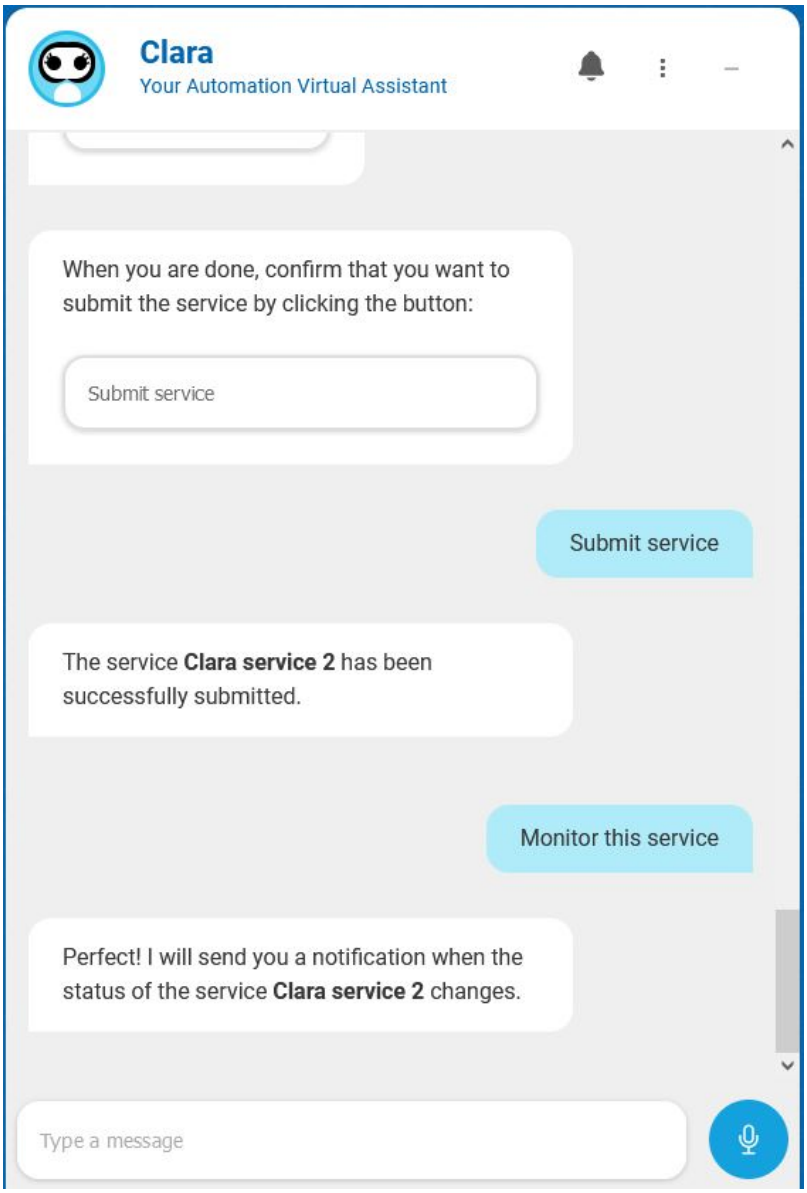



When you select this action, Clara shows the services available in the Self-service Catalog.




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 your custom values. Select a parameter to set or change its value. A link to Self-Service Catalog is provided if you want to  modify the parameters and submit the service from there.
4. Click **Submit service.**
5. Click the suggested action **Monitor this service** to ask Clara to monitor the service execution for you and send a **notification** when the service status changes.
   **Note**: Remember to enable Clara to send notifications. If you don't enable 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 Workload Automation servers through **HERO (HEalthcheck & Runbook Optimizer)**. This feature is available for WAd users only (not WAz).
With Clara, you can monitor one environment at a time.

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 documentation.
Before using this feature, you must properly configure Clara. For details, see HERO integration.

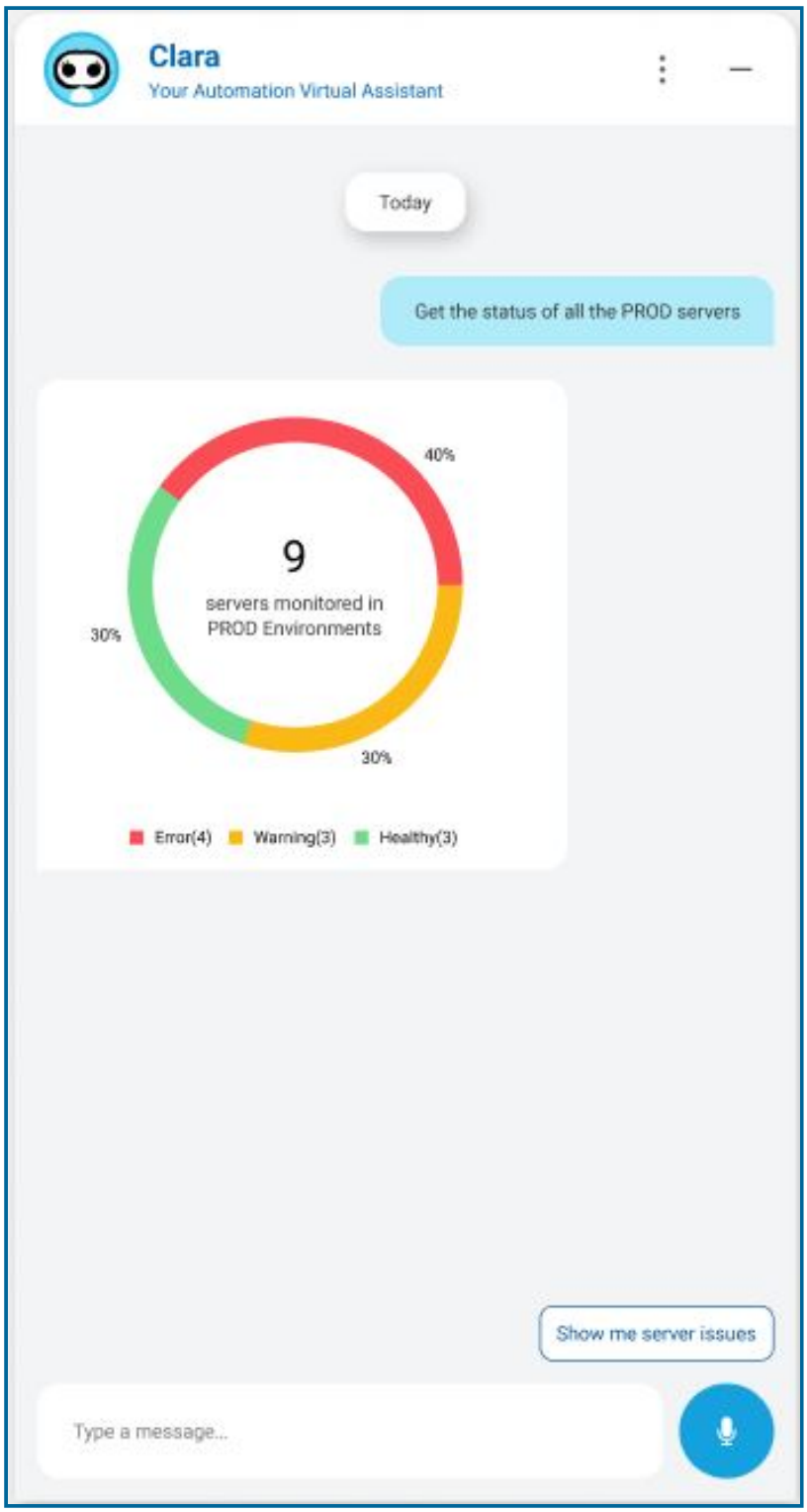
From the *What can you do?* conversation flow, select **Monitoring with HERO.**

The following actions are available:

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

## Get the status of all the PROD servers

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




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

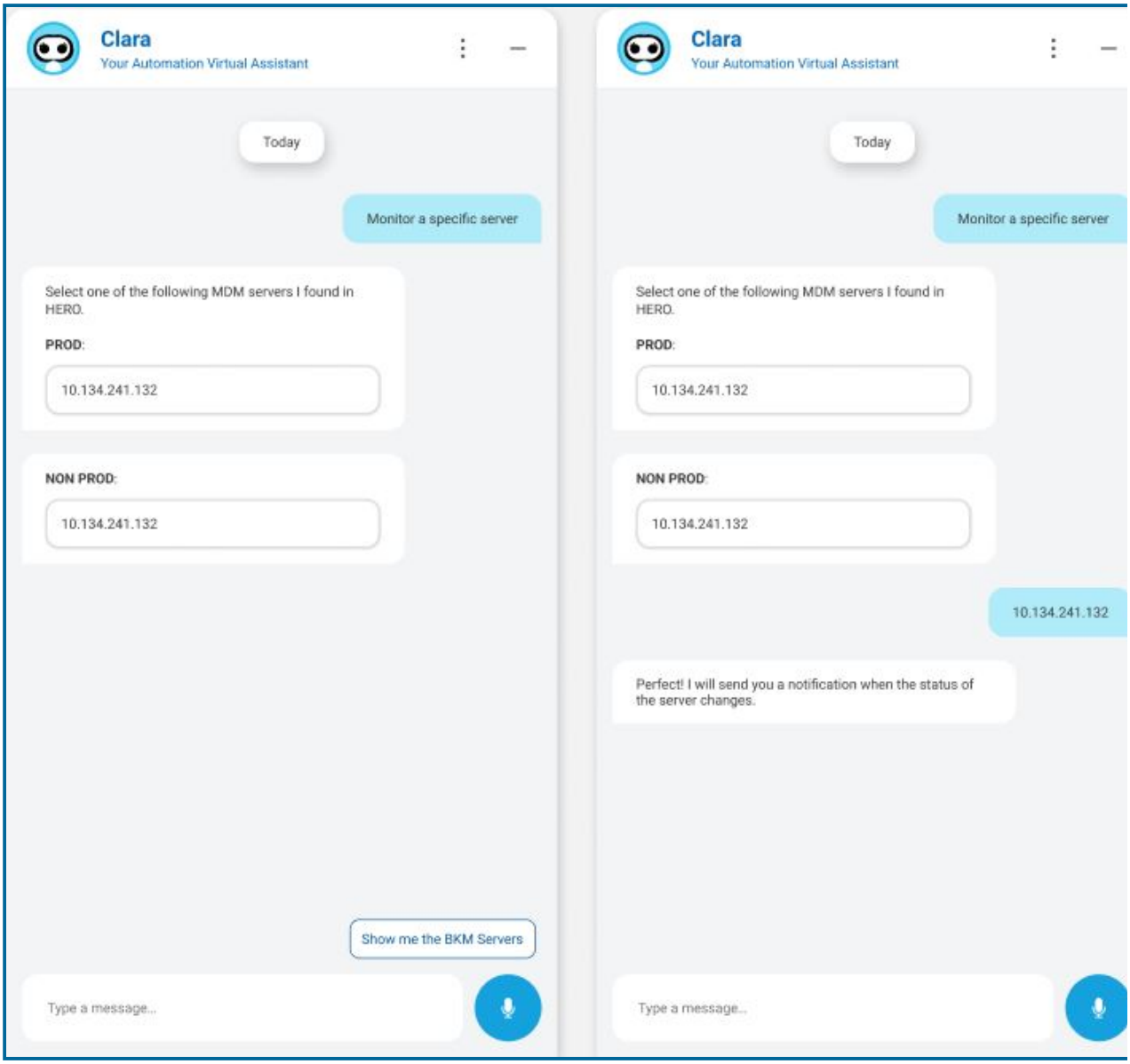
- Healthy
- Warning
- Error

If there are servers in **error** or **warning** status, click the suggested action **Show me server issues** to get the list of your **production** MDM servers in error or warning status.

Click the suggested action **Show me the BKM servers** to extend your search to BKM servers.
Select a server in the list. Clara provides details about the issue, such as a description, the failing component(s) and the environment type.
A link to HERO console is also available if you want to check the server directly in HERO and get additional information.
Click the suggested action **Monitor this server** to be notified when the server status changes.

## Monitor a specific server

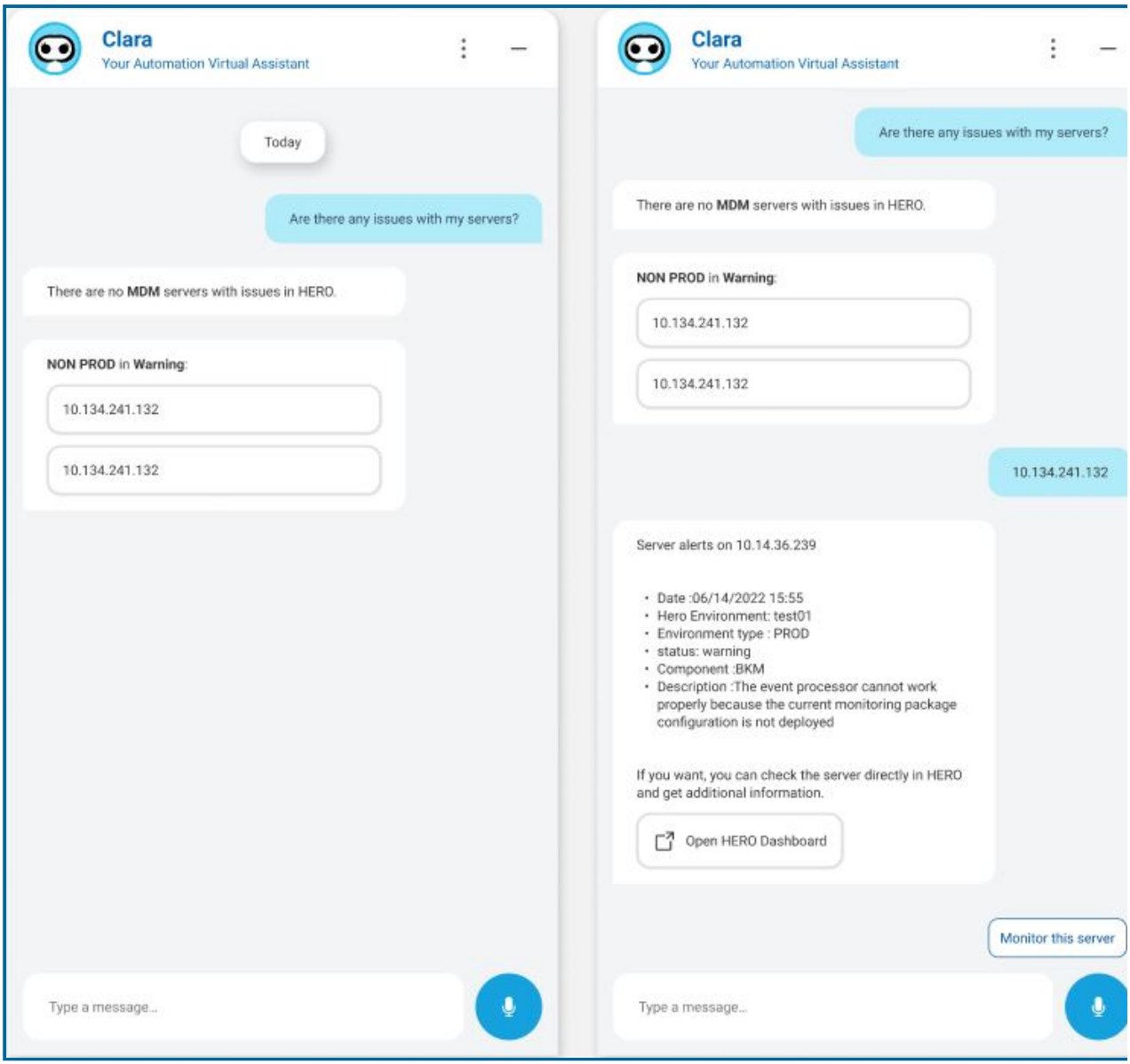
By selecting this action, you ask Clara to monitor a specific server on your behalf, and send you a **notification** when the server status changes.



Clara provides the list of your **production** and **non production** MDM servers.
Click the suggested action **Show me the BKM servers** to extend your search to BKM servers.
Select a server in the list. Clara starts monitoring the status of the selected server and sends you a notification when the status of the server changes.

## Are there any issues with my servers?

By selecting this action, you ask Clara to quickly discover if there are any issues (errors or warnings) with your **production** and **non production** servers (MDM and BKM).

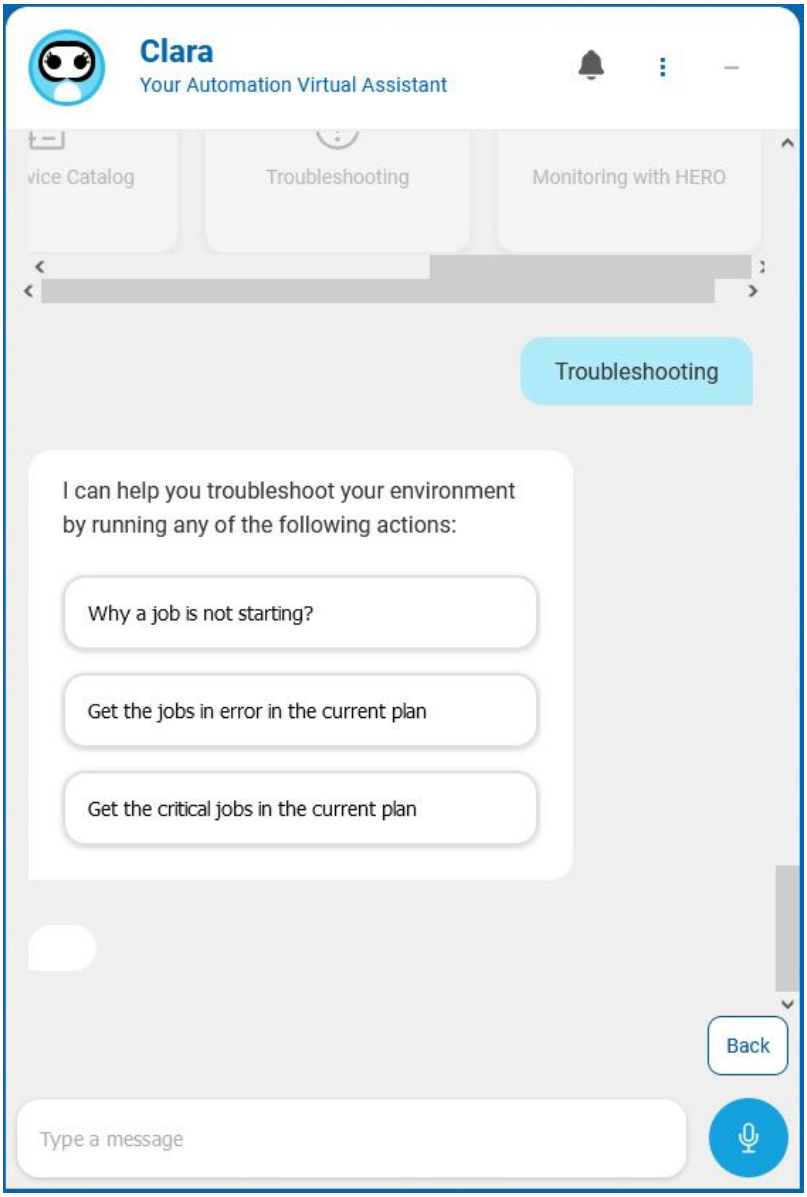


Clara provides the list of your **production** and **non production** MDM servers in error or warning status.
Click the suggested action **Show me the BKM servers** to extend your search to BKM servers.
Select a server in the list. Clara provides details about the issue, such as a description, the failing component(s) and the environment type.
A link to HERO console is also available if you want to check the server directly in HERO and get additional information.
Click the suggested action **Monitor this server** to be notified when the server status changes.

# Clara helps in problem determination

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

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




The following options are available:

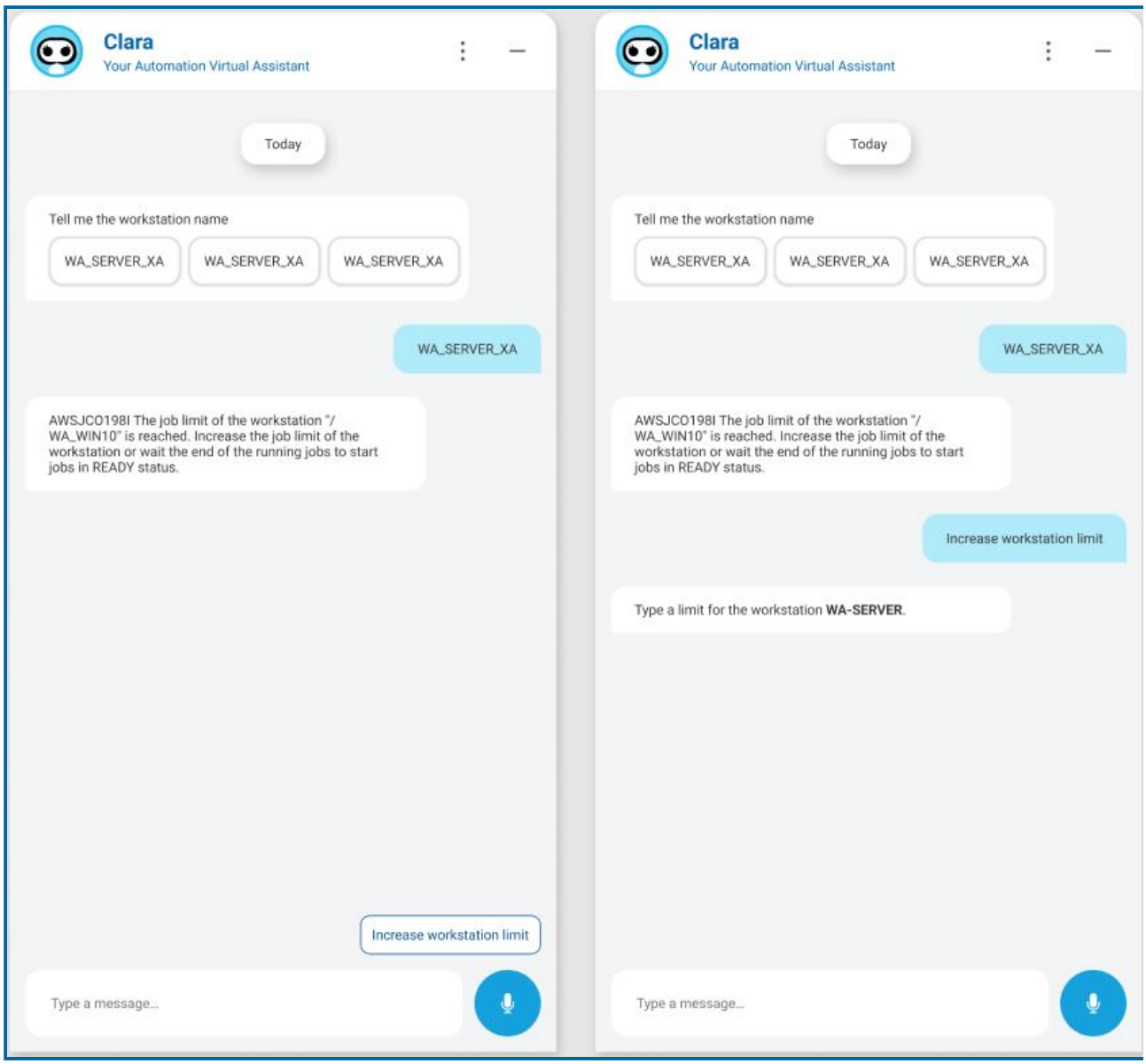
- **Why a job is not starting?** - You can find out why a job is not starting (available for WAd only).
- **Get the jobs in error in the current plan** - You can find all the jobs in error in the current plan (available for WAd and WAz).
- **Get the critical jobs in the current plan** - You can find all the jobs in high or potential risk of missing their deadline in the current plan (available for WAd and WAz).

## Why a job is not starting?

After selecting this action, provide the job name or select one from the list of jobs supplied by Clara.
Clara can understand if a job doesn’t start for any of the following reasons:

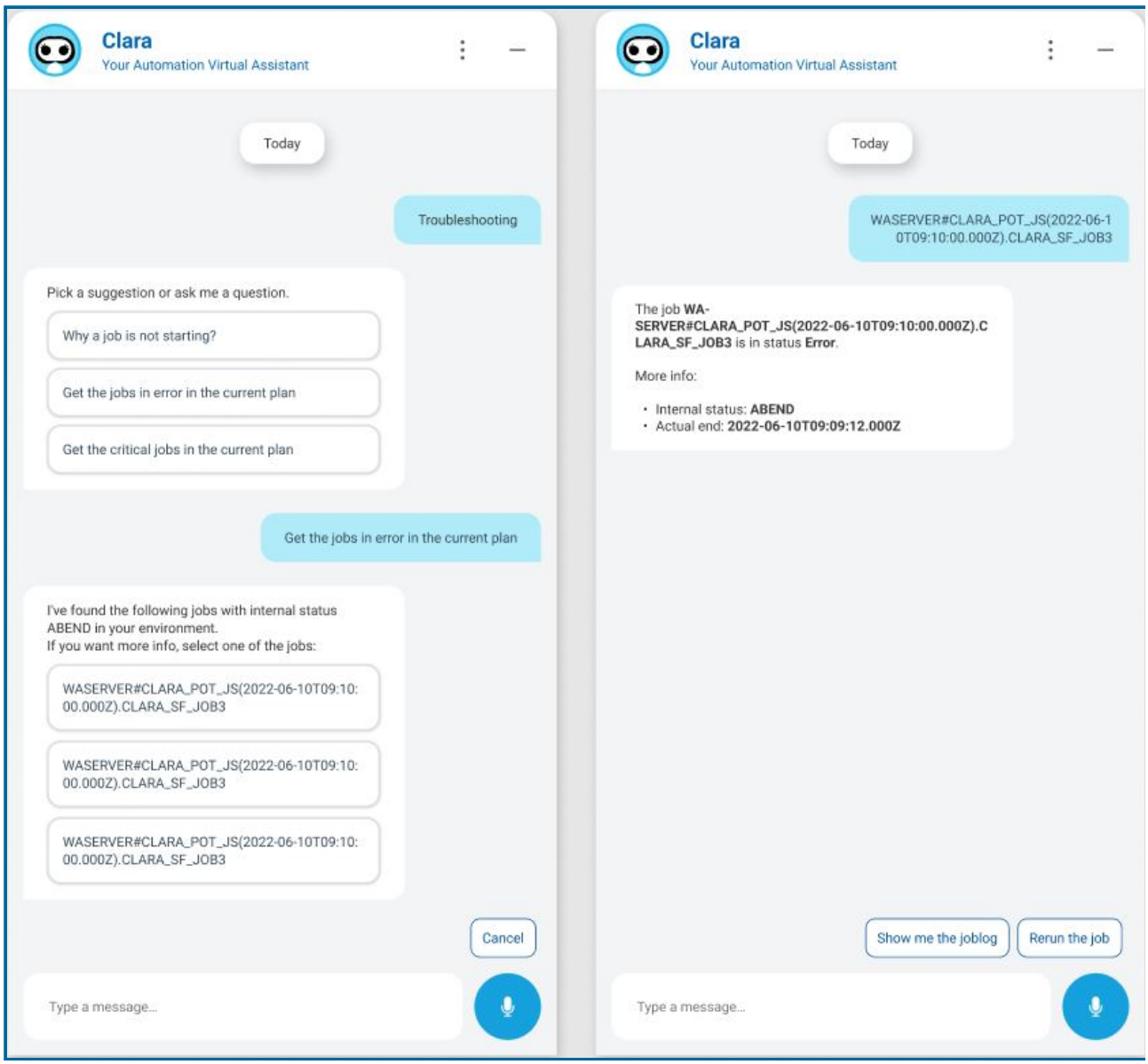
- the job limit of the workstation is set to zero
- the agent is unlinked, or stopped
- there are unresolved dependencies
- the job priority is set to zero

In all the above situations, Clara suggests the next action she can run to solve the issue, such as changing the priority, increasing the limit, or releasing the dependencies.



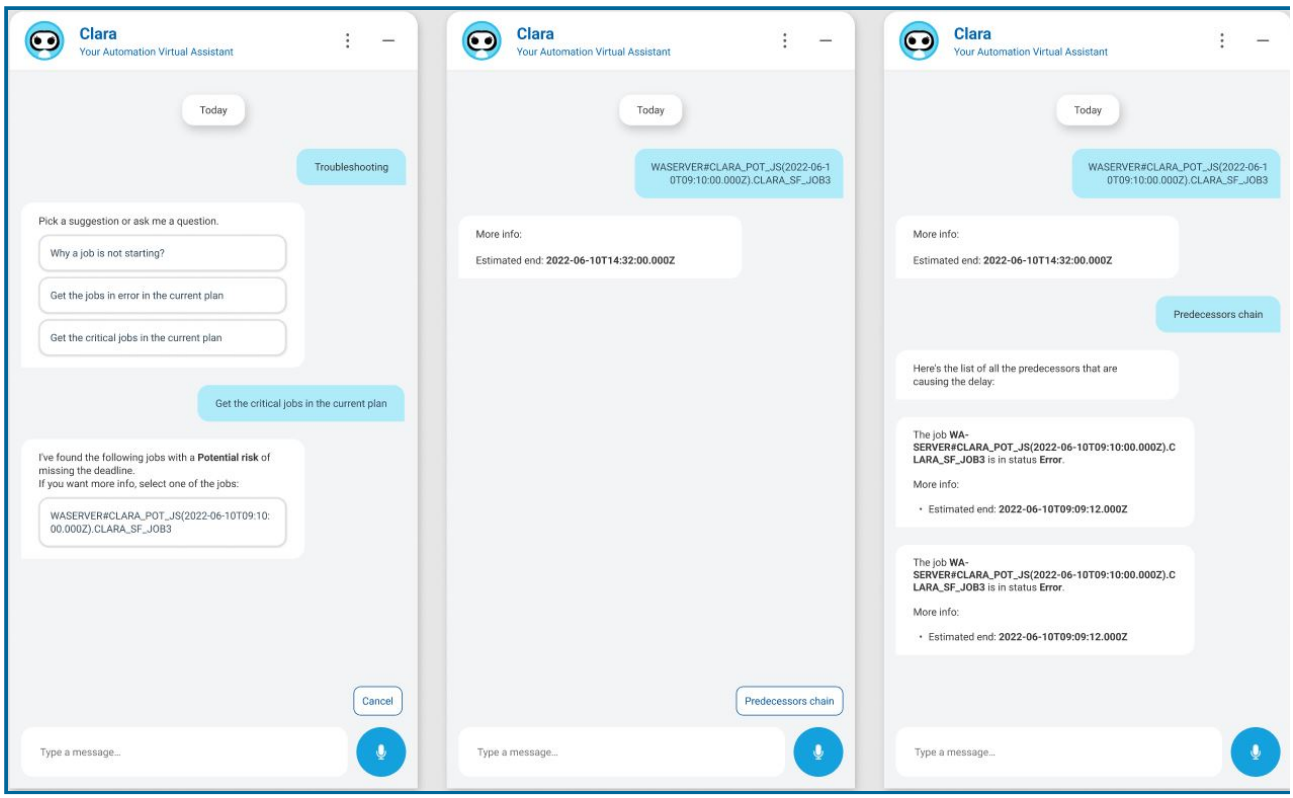

## Get the jobs in error in the current plan

By selecting this action, Clara displays the list of jobs in error in the current plan.
Select a job from the list. Clara provides more information about the job and suggests, as a next action, to analyze the joblog or rerun the job.




## Get the critical jobs in the current plan

By selecting this action, Clara displays the list of jobs in **high** or **potential risk** of missing their deadline in the current plan.
Select a job from the list. Clara provides more information about the job and suggests, as a next action, to display the list of predecessor jobs that are causing the delay.



# 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

Clara is built adopting a microservices-based architecture. 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:
    - For reverse proxy container:
      $(docker logs clara-nginx -t) > clara-nginx.log
    - For identity management container:
      $(docker logs clara-keycloak -t) > clara-keycloak.log
    - For credential manager container:
      $(docker logs clara-manager -t) > clara-manager.log
    - For Clara-nlg container, providing Clara response pack:
      $(docker logs clara-nlg -t) > clara-nlg.log
    - For Clara-core container, providing Clara learning models:
      $(docker logs clara-core -t) > clara-core.log
    - For Clara-actions container, providing all actions for Workload Automation rest APIs:
      $(docker logs clara-actions -t) > clara-actions.log
    - For Opensearch container:
      $(docker logs clara-datastore -t) > clara-datastore.log
    - For Apache Solr container:
      $(docker logs clara-search-engine -t) > clara-search-engine.log
    - For Clara asynchronous service container (Clara asynchronous service handles the execution of asynchronous task interacting with WA):
      $(docker logs clara-asyncservice -t) > clara-asyncservice.log
    - 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-manager -t) > clara-manager.log ; $ (docker logs clara-nlg -t) > clara-nlg.log ; $(docker logs clara-core -t) > clara-core.log ; $(docker logs clara-actions -t) > clara-actions.log ; $(docker logs clara-datastore -t) > clara-datastore.log ; $(docker logs clara-search-engine -t) > clara-search-engine.log ; $(docker logs clara-asyncservice -t) > clara-asyncservice.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 nginx".

**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**

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

**Cause and solution**

Add an exception to trust the website.

**Symptom**

ElasticSearch does not start, and the log shows the following error:
"2018-09-21T14:11:16,039][INFO ][o.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 deamon) 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 65636, 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 deamon:
   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**

Run the following commands:

./clara.sh --down --vol

./clara.sh --up

**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 HCL_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:

./install.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.

**Symptom**

Selecting **Change Language** from the menu in the upper right corner of Clara's interface, Clara replies with the following message:

"You cannot change language, translation is not available".

**Cause and solution**

- The Translator service has not been configured. See Installing and configuring the Translator service.
- The Translator service cannot be connected. If the Translator service is installed on a different machine, check it is reachable from Clara's machine.

**Symptom**

When asking Clara to manage workload from the Self-Service Catalog, Clara replies with the following message:
"I can't find any services assigned to your role".

**Cause and solution**

Verify if the Workload Automation administrator has correctly authorized users to catalog and services. For details, see the chapter *Authorizing users to access catalog and services* in the Mobile's Applications User's Guide of Workload Automation.