HCL HERO V1.0.0.9

HCL HERO documentation

Welcome to the **HCL HERO (HEalthcheck & Runbook Optimizer)** documentation, where you can find information about how to install, configure, and use HERO.

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. HERO brings the benefits of Artificial Intelligence (AI) technology to customers. By providing both actual KPIs (such as throughput and queue monitoring) and AI-powered trend estimation of KPIs, HERO enables the prediction of potential problems.

For information about **HCL Workload Automation**, see HCL Workload Automation documentation.

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

Product Overview

Enterprises depend on **Workload Automation** to manage business critical workloads, reduce operating costs, and deploy new services faster. **HCL HERO** effectively helps IT administrators monitor the health of their servers and perform informed recovery actions with specialized Runbooks, keeping your multiple Workload Automation environments responsive and reliable. It also helps reduce manual labor, reduce downtime of servers, and improve IT operational efficiency across the enterprise.

Capabilities

HERO effectively combines centralized application monitoring with Runbook automation, providing a solution dedicated to Workload Automation, that makes your infrastructure easy to control and hassle free.
It provides the following capabilities:

- Monitor the health of multiple production and non-production environments simultaneously
- Easily integrate a Runbook library with customized monitors and KPIs
- Provide dedicated performance indicators for Workload Automation, with predictive analysis and Runbook recommendation
- Recommend and execute Runbooks correlated to specific issues identified on the components of your application server infrastructure

Benefits

HERO brings the benefits of Artificial Intelligence (AI) technology to customers. By providing an automated, intelligent solution to monitor the health of your Workload Automation environments, HERO frees up IT Administrator’s time and reduces manual effort:

- Out-of-the-box monitors and Runbooks
  - Eliminate the need to manually maintain custom monitor scripts
  - Open platform to integrate predefined custom monitor scripts and Runbooks
- KPIs trends prediction
  - Provide actual KPIs, such as throughput and queue monitoring
  - Provide AI-powered trend estimation of KPIs to predict potential problems
- Failure prevention
  - Provide intuitive dashboard with warnings, errors prioritization, and email notification
  - Control internal queues exhaustion before system hangs
- Achieve optimum performance
  - Get insights on your infrastructure weaknesses
  - Plan resources allocation

What's New

See what's new with HERO releases.

HERO V1.0.0.9

HERO V1.0.0.9 provides the following enhancements:

1. Simplified Installation
   Enhanced setup process designed for ease of use and accelerated deployment.
2. Pre-Built Image
   Pre-configured container images now available, significantly reducing installation time.
3. Rootless Installation Support
   Introduced support for rootless installations, enabling secure and flexible operation without requiring root privileges.
4. Podman Container Engine Compatibility
   Added compatibility with Podman, a daemonless container engine, providing an alternative to Docker.
5. Improved Log File Management
   Optimized container log management to efficiently conserve disk space on the host system.
6. Direct Access to Official Documentation
   New option to access official Hero documentation directly from the Hero home page, enhancing user convenience.

HERO V1.0.0.8

HERO V1.0.0.8 provides the following enhancements:

1. Support of Workload Automation V10.2.0 and V10.2.1
2. Flexera license check removed
3. Podman V3 support added as a container engine to support RHEL
4. Installation improvement:
   - Easier installation procedure
   - Master Licence Agreement displayed to users for acceptance

HERO V1.0.0.7

HERO V1.0.0.7 provides the following enhancements:

1. Enhancements of the Log Parsing feature
   - New predefined log parsing rules. For details, see Log Parsing and Alert Setting.

2. Enhancements of HERO User Interface:
    - New button on the KPIs page to view KPIs trend in zoom-out mode.
3. Migration from Elasticsearch and Kibana to OpenSearch and OpenSearch Dashboards.

HERO V1.0.0.6

HERO V1.0.0.6 includes the following enhancements:

1. Support of Workload Automation V10.1.0.

HERO V1.0.0.5

HERO V1.0.0.5 includes the following enhancements:

1. Workload Automation database connection through JDBC driver is no longer required. With HERO V1.0.0.5, you can get information about throughput and critical agents by using Workload Automation APIs.
2. Prediction improvement
3. HERO User Interface has been renewed to improve the user experience.

HERO V1.0.0.4

HERO V1.0.0.4 includes the following enhancements:

1. Creation, management, and monitoring of product log parsing rules with alert generation. See Log Parsing and Alert Setting.
2. Support of SSL encryption for JDBC connection.

HERO V1.0.0.3

HERO V1.0.0.3 includes the following enhancements:

1. Five additional Monitors
    - Final Job Stream Check
    - Event Processor Monitor
    - Broker Process Status

    - WAS Certificate Expiration Check
    - DB2 Certificate Expiration Check
2. Two additional Runbooks
    - Check rule status
    - Start event processor
3. An additional KPI collector
    - Job trends

Product Videos

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

HERO for Workload Automation

Automate more, better, and smarter with HCL Automation Power Suite

How to install HCL HERO V1.0.0.3

Customer Support

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

About the Installation Guide

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

System Requirements

HERO is installed using Docker.
You can install HERO on Linux 64 bit operating system.

Supported browsers

- Google Chrome
- Mozilla Firefox (Quantum recommended)

Software requirements

- Docker CE (Community Edition) Engine 18.09.0 or later

- Docker Compose 1.23.2 or later

Hardware requirements

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

- CPUs 4
- RAM 24GB
- Swap 1024MB

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

- CPU 64 bit, 2+ core
- RAM 32GB+
- Storage 200 GB HDD

Workload Automation Requirements

HERO can monitor Workload Automation environments starting from Workload Automation version 9.1. in which the prerequisite feature **Plan data replication in the database** (also known as "mirroring feature") is enabled. For more information about the “mirroring” feature, see Replicating plan data in the database.

HERO supports direct monitoring for Master Domain Manager (MDM), Backup Domain Manager (BKM), and Dynamic Workload Console (DWC). Monitoring for all the remaining agents is performed using the database on MDM.

The supported operating systems for MDM, BKM, and DWC are **AIX** and **Linux 64 bit.**

Prerequisites for connecting the monitored environments

Once the monitoring scripts are deployed to the target workstations, they are scheduled by HERO to run on a regular basis.

The HERO server requires a connection to each DB2 or Oracle instance in the monitored environments from which to collect throughput data and information about the agent status.

The following prerequisites must be met for HERO to monitor Workload Automation environments:

- SSH server daemon must be up (SSH daemon allows authentication with username and password)
- The user specified at workstation discovery time, i.e. the owner of the WA instance, must have the following permissions:
  - Permission to write in his own home directory and in the <TWA_HOME> directory. Access permission must be through SSH.
  - Permission to read the WA registries.
  - Permission to create job and job stream definitions, to submit, and cancel them.
- For some recovery actions, and for some monitors based on previous versions of Workload Automation, it might be necessary to run actions as “sudo”. In this case, the sudo user must be authorized without specifying a password.
- It must be possible for the target workstation to open an https or http connection with the HERO server (typically using port 8080, but the port number can be customized).

Starting from version 9.5, Workload Automation can be deployed on Docker containers. During the server discovery process, if containerized components are present on a server, you are requested to provide the image name for each component to be retrieved.

Ports to be opened for the communication between HERO and the product servers

For the communication between HERO and the product servers, make sure ports are opened as follows:

- From HERO to the product servers:
  - SSH port (default port 22), used by HERO to connect to the product servers
- From the product servers to HERO:
  - https on port 443 (default port) or any custom port, required for status updating
- For machines connected through WinRM protocol over https, port 5986 must be opened.

System requirements and considerations for training and prediction

Hardware requirements depend on the number of WA servers where you want to run the prediction, but you must consider that CPUs with AVX instructions support are required (for this reason, old CPUs might not work).

To start the training process, it is recommended to have at least 10 hours of new data for the prediction procedure.

The performance, in terms of computational time of the training (or retraining) procedure, is highly dependent on the technical specifications of the machine on which it runs.

There are no minimal requirements with the exception of 4GB of RAM to run FBProphet.
With 8 GB of RAM available and an Intel core i7 processor, the estimated time is about 12 minutes for the training phase per server.
In addition to this time, the swap in / out time for writing and reading from OpenSearch should also be considered.

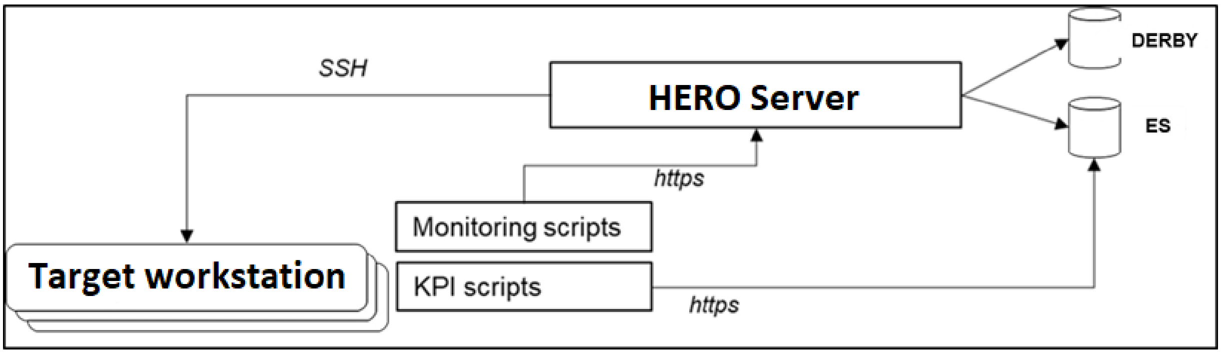
Basic Architecture

HERO consists of three main components:

- HERO Server
- HERO Historical Database (OpenSearch)
- HERO Transactional Database (Derby stored on the filesystem)

The HERO Server consists of two main components:

- HERO Web Application services (WAR application deployed on the Tomcat application server)
- HERO Dashboarding services (OpenSearch Dashboards)



To connect to the target workstations and run discovery operations and scripts, the HERO server uses the SSH protocol.
When a target workstation is discovered, the HERO server deploys a monitoring script for each monitor previously selected by the user. The scripts are deployed in the directory specified in the scriptPath property of the dashboard.property configuration file (default directory: <%USER_HOME%\HERO>), and are scheduled to run on a regular basis.

Starting from version 9.5, Workload Automation can be deployed on Docker containers. In this case, monitors run in the container where the monitored component has been deployed.
KPI Scripts collect Performance Indicators data and send it to the historical database.

HERO Quick Start Guide

A Quick Start Guide for HERO is available in PDF format. This guide describes a quick and easy way to install the product.
To download or open HERO Quick Start Guide, see Documentation in PDF format.

Installing and Configuring HERO

Checking system prerequisites

Before starting to install HERO, 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. HERO 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 HERO on RHEL or CentOS distros, SELinux must be set to Permissive or Disabled. See: How to set SELinux to permissive.
5. Verify that the Workload Automation user entitled to discover servers in HERO has access to the **crontab** command.

Installation procedure

To install and configure HERO, run the following procedure.

1. From HCL License Portal download the appropriate HERO installation package.
   - HCL HERO V1.0.0.9
     Hero is available with 2 different types of installation packages:
   - Installation package for **offline installation**:
     HCL_HERO_1.0.0.9_OFFLINE.tar.gz containing:
     - hero.tar.gz

    - hero-images.tar
    - install.sh
    - cert
  - Installation package for **online installation**: HCL_HERO_1.0.0.9_ONLINE.tar.gz containing:
    - hero.tar.gz
    - install.sh
    - cert

      The **online** installation requires a working internet connection to reach the Entitled Registry that contains Hero images. Contact your HCL sales representative for the login details required to access the Entitled Registry.

      Execute the following command to log into the Entitled Registry:

      ```
      docker login -u <your_username> -p <your_entitled_key> hclcr.io
      ```

      Hero images are as follows:
    - hclcr.io/slt/hero/hero_nginx :1.0.0.9
    - hclcr.io/slt/hero/hero_tomcat :1.0.0.9
    - hclcr.io/slt/hero/hero_configurer:1.0.0.9
    - hclcr.io/slt/hero/hero_elasticsearch:1.0.0.9
    - hclcr.io/slt/hero/hero_kibana:1.0.0.9
    - hclcr.io/slt/hero/hero-predictor:1.0.0.9
    - hclcr.io/slt/hero/hero_keycloak:1.0.0.9
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. To install Hero using a custom SSL certificate instead of the default one, follow these steps:
   i. Navigate to the <BUILD-DIR>\cert folder.
   ii. Place the custom SSL certificate files (hero.key and hero.cert) in cert directory.
   Note: Ensure that your custom files are named hero.key for the private key and hero.cert for the certificate.
4. Open a bash shell and get ready to install Hero.
   - From the <BUILD_DIR> directory, run the command:
     Note: This function installs Hero on the target machine by utilizing the files and folder structure extracted from the provided shipping package.
     By default, before starting the installation, Hero checks the installation

prerequisites. If you want to skip this check, add the --skip-check-prereq parameter. For details, see the install.sh command usage.

./installHERO.sh --host <hostname> [--skip-license-check, --port <port>, --install-path <install path>, --prune-docker-pv, --skip-check-prereq, --podman, --client-secret]

Parameter	Description	Mandatory
--host <hostname>	The hostname or IP address of the machine on which you are installing Hero.	Yes
--port <port>	The network port where you are installing Hero (default port is 443).	No
--install-path <install path>	The path where you want to install Hero (default value is /opt/hcl).	No
--skip-license-check	Use this option to skip the license check.	No
--skip-check-prereq	Use this option to skip the pre-requisites check.	No
--prune-docker-pv	Use this option if you want to prune the docker volumes.	No
--podman	Use this option for installation with the Podman container engine.	No
--client-secret	The client secret for the keycloak.	Yes

5. The installation script dispalys the HCL Master License Agreement (MLA) for your acceptance. Scroll down or press the "q" key to quit. Type "Y" to accept the MLA.
6. The installation script runs the installation process, and verifies its successful completion.

Post installation steps

When the installation is complete, the following link is prompted to login to the HERO dashboard: **https://<your_host_machine_address:port>/Dashboard**

The installation script creates two default HERO user accounts:

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

Note:

At login time, if you enter wrong credentials, after 5 unsuccessful attempts you will be locked out for 30 minutes.

After completing the installation successfully, you can proceed to configure email alerts. Use the Keycloak administration console to define new users, new roles, or change default passwords. You can access Keycloak administration console at the following link: https://<IP:PORT>/keycloak/auth/admin by using the following credentials:

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

If you want, you can change Keycloak default password. For instructions, see Configuring Security.

Before adding environments to the HERO dashboard, verify that the installation process has created **OpenSearch Dashboards default index pattern:**

- In the environment page, check if the KPI link appears on top of the server card.
- If the link doesn't show up, manually set the OpenSearch Dashboards default index pattern by following the procedure in the Appendix.

To stop HERO (for example, after changing some configuration parameters), from the <install-path> directory, type **docker-compose stop.**
To restart HERO, from the <install-path> directory, type **docker-compose start.**

For any reference, you can find the manual installation procedure in the Appendix. It guides you to manually execute the steps run by the automatic installation script.

Managing containers

To manage HERO containers, run the following procedures from the <install-path> directory.

- To gracefully stop/restart HERO, for example after reconfiguring HERO, run the following commands:
  1. **docker-compose stop**
  2. **docker-compose start**
- To reset the containers, while maintaining HERO configuration and data stored in the DB, run the following procedure:
  1. **docker-compose down**
  2. **docker-compose up --build -d**

  This procedure doesn't reset the custom client secret if you have created one (see Configuring Security).
- To reset the containers and the HERO configuration, while maintaining the data stored in the DB, run the following procedure:
  1. **docker-compose down**

2. **docker volume rm <install-path>_hero-home**
3. **docker-compose up --build -d**

  This procedure doesn't reset the custom client secret if you have created one (see Configuring Security).

- To reset the containers, the HERO configuration, and the data stored in the DB, run the following procedure:
  1. **docker-compose down**
  2. **docker volume rm <install-path>_hero-home <install-path>_ build_hero-db-data <install-path>_ build_hero-es-data <install-path>_ build_keycloak-nginx-ssl <install-path>_ build_pgdata**
  3. **docker-compose up --build -d**

  This procedure resets also the custom client secret if you have created one (see Configuring Security). Reconfigure HERO with the default client secret or create a new one.
- In addition, to delete also HERO images, run the following command:
  docker rmi <install-path>] _tomcat <install-path>_prediction <install-path>_keycloak <install-path>_nginx <install-path>_kibana docker.elastic.co/kibana/kibana-oss docker.elastic.co/elasticsearch/elasticsearch-oss
  This command completely uninstall HERO.

Applying changes to Runbooks and Monitors

Every time a runbook is added, or a new monitor is created, run the following commands from the <install-path> directory:

1. **docker stop hero-tomcat**
2. **docker rm hero-tomcat**
3. **docker volume rm <install-path>_hero-home** (to remove the configuration volume)
4. **docker-compose up --build -d**

Configuration Files

The **dashboard.properties** file contains general configuration parameters:

disk-space	Minimum percentage for the disk space monitor to generate an alert.
IPdashboard	The URL of the HERO server used by the monitoring scripts.

IPdashboard_curl_options	The options used by the CURL command run by the monitoring scripts. Used for authentication purposes on the HERO server.
elasticsearch_external	The historical database (OpenSearch) URL to be used by the monitoring scripts running on the workstation.
elasticsearch_curl_options	The options for the CURL command run by the KPI scripts. Used for authentication purposes on OpenSearch.
Queue_< queue name >_limit	Warning that notifies when the queue availability is lower than the limit that you set. Supported only for Workload Automation.
esClientLink	The link used by the HERO server to reach the Historical Database (OpenSearch).
kibanaLink	Link to OpenSearch Dashboards.
scheduledTime	The frequency of the scheduling operations for HERO monitors.
esQueueMapping	Template for creating the index on OpenSearch.
esQueueDashboardTemplate	Template for creating the queue dashboard.
esQueueChartVisualizationTemplate	Template for creating the chart visualization for the queue.
esQueueGaugeVisualizationTemplate	Template for creating the Gauge visualization for the chart in the dashboard.
esQueuePanelJSONTemplate	Template for creating the Queue panel.
esThroughputMapping	Template for creating the OpenSearch index for throughput.
esThroughputChartVisualizationTemplate	Template for the throughput visualization for the chart in the dashboard.
esThroughputPanelJSONTemplate	Template for creating the throughput panel.
LaunchInContextUrl	The url of the HERO UI that will be used in the alert emails.

tempPath	The path on the HERO Server where the monitor files will be stored before deploying.
runbookLimit	Maximum number of visualized runbooks.
alertSmtpEmail	Optional if you want to configure alerting by email. The sender email account [Example: username@gmail.com].
alertSmtpPassword	Optional if you want to configure alerting by email. The password associated to the sender email account.
smtpPasswordEncrypted	Optional if you want to configure alerting by email. Set encryption for the alertSmtpPassword. Can be "true" or "false". If smtp is configured through HERO installation script, the value of smtpPasswordEncrypted parameter is set to "true" (default value). If you configure smtp manually, you must set smtpPasswordEncrypted to "false".
alertSmtpServer	Optional if you want to configure alerting by email. Fully qualified hostname of the SMTP Server that will be used by HERO to send alerts by email [Example: smtp.gmail.com].
alertSmtpPort	Optional if you want to configure alerting by email. The port of the SMTP mail server.
smtpTlsEnabled	Optional if you want to configure alerting by email. Set the TLS enablement for smtp client while establishing a connection from HERO. Can be "true" or "false". Default value is "true".
sslTrustStorePassword	Optional if you are using a custom SSL truststore file for your JDBC connection, you are required to provide an SSLtruststore password.
maxLogsShown	Maximum number of visualized logs.

pwdNeedsEncryption	Require that alertSmtpPassword must be encrypted on the first execution of SMTP application.
taskSchedulerThreadPoolSize	Max number of threads for scheduled monitors.
predictionIndexMapping	Template to create the prediction index on OpenSearch.
deployPath	Home directory in which monitoring scripts are deployed.

The **ui.properties** file configures the connection to the HERO web application services:

ip	The hostname or IP address of the HERO server.
port	The port of the HERO server. Default value is 443.
kibanaHost	The hostname of the dashboarding service (OpenSearch Dashboards) that is reachable by the browser.
kibanaPort	The port of the dashboarding service (OpenSearch Dashboards) that is reachable by the browser.
protocol	The protocol to be used (http or https).
wsProtocol	The protocol used for the shell inside HERO, this protocol should be ws if the protocol property is http, otherwise this should be wss.
sshPort	The ssh port for the connection to other machines (usually 22).
keycloak	Used to configure the connection to Keycloak. Set this variable to the same value at which you set the Keycloak_URL parameter in the .tomcat.env file. For details, see Configuring Security.
roles	Available roles in the HERO Keycloak security configuration, separated by comma.

clientSecret	Used to configure HERO with a new secre in place of the default one. For details, see Configuring Security.

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

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

Upgrading HERO

You can upgrade your HERO installation from V1.0.0.6 or V1.0.0.7 or V1.0.0.8 to V1.0.0.9. If you start from previous versions, you can only get a fresh installation of V1.0.0.9.

If upgrading from V1.0.0.8 to V1.0.0.9, run the following procedure:

1. Backup the existing OpenSearch data.
   Execute the following commands to export data from the previous OpenSearch version:
   ```
   docker cp hero-elasticsearch:/usr/share/opensearch/data ./
   ```
   Note: After running the command, a data folder will appear in the current working directory.
2. From Keycloak administration console, in the left side navigation bar, select **Clients > nginx**. From the tab **Credentials**, copy the content of the field **Secret** and save it.
3. Clean Up Old Hero Installation Containers:
   - Navigate to the hero-1.0.0.8 directory.
   - Run the following command to stop and remove the existing containers:
   ```
   docker-compose down
   ```
4. Install Hero 1.0.0.9:
   i. Extract the HERO v1.0.0.9 package into a new directory, distinct from the HERO v1.0.0.8 installation directory.
   ii. Launch the simplified installation process for HERO v1.0.0.9. Refer to the Installing and Configuring HERO documentation for detailed steps.
   Note: Provide the client secret retrieved in the previous step as a command-line argument during installation.
   bash
5. After completing the installation successfully, perform the following steps to migrate data from Old Persistent Volumes to New Persistent Volumes:

```
docker run --rm -v <BUILD_DIR>_hero-data:/from -v hero_hero-data:/to alpine sh -c "cp -a /from/. /to/"
docker run --rm -v <BUILD_DIR>_hero-db-data:/from -v hero_hero-db-data:/to alpine sh -c "cp -a /from/. /to/"
docker run --rm -v <BUILD_DIR>_hero-es-data:/from -v hero_hero-es-data:/to alpine sh -c "cp -a /from/. /to/"
docker run --rm -v <BUILD_DIR>_hero-home:/from -v hero_hero-home:/to alpine sh -c "cp -a /from/. /to/"
docker run --rm -v <BUILD_DIR>_keycloak-data:/from -v hero_keycloak-data:/to alpine sh -c "cp -a /from/. /to/"
```

6. Import the Backup OpenSearch Data and Set Appropriate Permissions:
   i. Copy the backed-up data to the OpenSearch container using the following command:
   ```
   docker cp ./data hero-elasticsearch:/usr/share/opensearch/
   ```

   Note: Execute this command from the directory where the backed-up data (created in Step 1) is located.

   ii. Verify that the data has been successfully transferred and ensure appropriate permissions are applied within the container, if necessary.
7. Grant Required Permissions on Bind Persistent Volumes (PVs).
   For OpenSearch container:
   i. Access the OpenSearch container with root privileges:
   ```
   docker exec -u root -it hero-elasticsearch /bin/bash
   ```
   ii. Change ownership of the OpenSearch data directory to the opensearch user:
   ```
   chown -R opensearch:opensearch /usr/share/opensearch/data
   ```
   iii. Set the appropriate permissions for the data directory:
   ```
   chmod -R 755 /usr/share/opensearch/data
   ```
   iv. Exit the container:
   ```
   exit
   ```
   For Hero Tomcat container:
   i. Grant full read, write, and execute permissions on the database directory:
   ```
   chmod -R 777 /deployments/apache-tomcat/database
   ```
   ii. Access the hero-tomcat container with root privileges:
   ```
   docker exec -it --user root hero-tomcat sh
   ```
   iii. Exit the container:
   ```
   exit
   ```
8. Restart All Containers

i. Navigate to the Hero 1.0.0.9 installation directory (default is /opt/hcl/hero):

```
cd /opt/hcl/hero
```

ii. Restart all containers using Docker Compose:

```
docker-compose restart
```

If upgrading from V1.0.0.6 or V1.0.0.7, run the following procedure:

1. From Keycloak administration console, in the left side navigation bar, select **Clients > nginx**. From the tab **Credentials**, copy the content of the field **Secret** and save it.
2. Backup the content of the build folder of the previous HERO installation and keep track of the value assigned to the parameters asked in prompts by the installation script.
3. From <HERO_installation_dir> directory, run the command **rm -rf * & rm .tomcat.env**
4. Unzip V1.0.0.8 HERO package in the same <HERO_installation_dir> directory.
5. Edit the following files:

   ui.properties in <hero_home>/CONFIGURATION/HERO/
   .tomcat.env in <hero_home>/

   and replace the secret key with the value you saved in step 1.
6. When **upgrading from V1.0.0.6**, edit opensearch.yaml in <hero-home>/DEVOPS/elasticsearch/config/ and update path.data value to “/var/lib/elasticsearch”.
   When **upgrading from V1.0.0.7**, edit opensearch.yaml in <hero-home>/DEVOPS/elasticsearch/config/ and update path.data value to “/usr/share/opensearch/data”.
   Note: the value of path.data in opensearch.yml should be the same as the hero-es-data volume value in docker-compose.yaml present in <hero-home>.
7. Launch the installation (see Installing and Configuring HERO) and reply with 'n' when the installation procedure prompts you the question "Do you want to prune Hero volumes [y/n]:"
8. Log into HERO UI: for all the already existing HERO servers click the gear button and select **Refresh Server.**

Configuring Email Alerts Service

Configuring the Email Alerts Service allows you to set up automatic notifications for key events and system updates. By defining the alert criteria and specifying email recipients, you ensure that relevant stakeholders are promptly informed. This service can be customized to trigger notifications based on specific conditions, such as system errors, updates, or user activity, enhancing communication and operational efficiency.

To configure alerts, run the following steps:

1. To enable email alerts, add the following settings to the dashboard.properties file located in the <install-path>/CONFIGURATION/HERO directory.
   If you are using the default installation path, the file can be found at: /opt/hcl/hero/CONFIGURATION/HERO.

alertSmtpEmail	The sender email account [Example: username@gmail.com] .
alertSmtpPassword	The password associated to the sender email account.
smtpPasswordEncrypted	Set encryption for the alertSmtpPassword. Can be "true" or "false". If smtp is configured through HERO installation script, the value of smtpPasswordEncrypted parameter is set to "true" (default value). If you configure smtp manually, you must set smtpPasswordEncrypted to "false".
alertSmtpServer	Fully qualified hostname of the SMTP Server that will be used by HERO to send alerts by email [Example: smtp.gmail.com]
alertSmtpPort	The port of the SMTP mail server.
smtpTlsEnabled	Set the TLS enablement for smtp client while establishing a connection from HERO. Can be "true" or "false". Default value is "true".

2. Configure the target email in the Alert.js file located in the <install-path>/CONFIGURATION/HERO/Alerting directory.
   If you're using the default installation path, the file can be found at: /opt/hcl/hero/CONFIGURATION/HERO/Alerting.

   Update the Targets Field

   Add or modify the email addresses in the targets field as demonstrated below:

```
[
{

var alerts = [
    {
        "component" : "ENV",
        "state" : "ERROR",
        "action" : "SEND_EMAIL",
        "subject" : "Environment in error",
        "message" : "The environment %ENV_NAME% type %ENV_TYPE% is in error"
        "targets" : "email1@example.com;email2@example.com"
    },
    {
        "component" : "ENV",
        "state" : "WARNING",
        "action" : "SEND_EMAIL",
        "subject" : "Environment in warning",
        "message" : "The environment %ENV_NAME% type %ENV_TYPE% is in warni
        "targets" : "email1@example.com;email2@example.com"
    }
];
```

Replace email1@example.com and email2@example.com with the actual email addresses you want to use as the recipients.

3. **Restart Hero Docker Containers**

- Navigate to the <install-dir>/hero and execute the following commands to stop and start the Docker containers:

  ```
  docker-compose stop
  docker-compose start
  ```

  By default, the installation directory is:

  ```
  /opt/hcl/hero
  ```

  Note: Ensure that the docker-compose.yml file is located in the <install-dir>/hero directory.

Creating Custom Email Alerts

To enable alert generation, HERO uses a built-in alerting framework that can be customized with different logic. For example, you can decide that an alert is generated when one of the environments is in error or warning status.
The customization is managed by a JavaScript file in which your can develop the logic for triggering an alert, and the action that the alert must run like, for example, sending an email, or executing a command on the machine that generated the alert.

To configure alerts, run the following steps:

1. To enable alerts by email, add the following settings to the **dashboard.properties** file located in the <BUILD_DIR>\CONFIGURATION\HERO directory:

alertSmtpEmail	The sender email account [Example: username@gmail.com] .
alertSmtpPassword	The password associated to the sender email account.
smtpPasswordEncrypted	Set encryption for the alertSmtpPassword. Can be "true" or "false". If smtp is configured through HERO installation script, the value of smtpPasswordEncrypted parameter is set to "true" (default value). If you configure smtp manually, you must set smtpPasswordEncrypted to "false".
alertSmtpServer	Fully qualified hostname of the SMTP Server that will be used by HERO to send alerts by email [Example: smtp.gmail.com]
alertSmtpPort	The port of the SMTP mail server.
smtpTlsEnabled	Set the TLS enablement for smtp client while establishing a connection from HERO. Can be "true" or "false". Default value is "true".

2. You can implement the alert logic by properly setting the file **Alert.js** located in the <BUILD_DIR>\CONFIGURATION\HERO\Alerting. The file is composed of three sections:
   Alerts

doStandardAlerts function
doAlerts function

The Alerts section is a list of objects that define different kind of alerts and is used by the **doStandardAlert** function.
The **doStandardAlert** function checks the alerts. When it finds the alert matching with the just happened event, it returns the alert and the framework runs the action that has been defined for that alert.

Alerts structure:

```
[
{
      "component": "ENV",         //Only works with ENV,
      "state": "ERROR",   //The trigger state, this can be: ERROR, WARNING, or SUC
      "action": "SEND_EMAIL",    //Only works with SEND_EMAIL
      "subject": "Environment in error",  //the email subject
      "message": "message1",     //the email message
      "targets": user1@email.com;user2@email.com      // email addresses
},
{
      "component": "ENV",
      "state": "WARNING",
      "action": "SEND_EMAIL",
      "subject": "Environment in warning",
      "message": "message2",
      "targets": "user3@email.com"

},
{….
 ….
}
]
```

For each alert, the **message** element contains specific variables that are replaced automatically by the system. For example, by setting:

```
       "message": "The environment %ENV_NAME% type %ENV_TYPE% is in error"
```

The target user will receive an email containing the name and type of the environment in error.

The **doStandardAlerts** function has the following parameters:
componentType: the type of the component
oldStatus: the previous status of the component
newStatus: the current status of the component
Variables: an object containing a list of useful information for the alerts

3. You can optionally configure Alerts in an advanced way, by using the **doAlerts** function. The **doAlerts** function has the same parameters of the **doStandardAlerts** function. However, it is not used to return alert objects but to execute the alerts directly from the **.js** file. The **doAlerts** function exposes two actions:
   AlertManager.runAction(command, machineId) which runs the shell command on the machine identified by the machineId
   AlertManager.sendEmail(subject, targets, message) which sends an email to all the targets defined by its parameters, where:
   targets is a string with all the email addresses separated by semicolon: "user1@email.com; user2@gmail.com"
   subject is the subject of the email that will be sent
   message is the email content and can be written in html

Creating Runbooks

To create a Runbook, add a file with name **RunbookName.js** in the folder <install-path>\CONFIGURATION\HERO\runbook (spaces in the file name are not allowed).
The file content must be the following:

```
function getRunBook() {
    var returnValue = {};
    returnValue = {
        actionID: "RunbookName",
        author: "John Nash",
        name: "Cool RunBook",
        description: "This runbook is very cool,
        component: "component_name",
        summary: "This is a test runbook",
        commands: "a command %HOME_DIR%
                             with the first parameter = %PARAM1%,
                              the second one = %PARAM2%
                              and the last one= %PARAM1%__%PARAM1% - - %PARAM_LIST%
                              and that's it",
        os: "linux",
        rating: 0,
        docLink: "http://www.google.com",
        readonly:true,
        parameters: [
                        {
                        "name": "PARAM1",
                        "type": "string",
                        "value":"",
                        "default":"my_name"
                        },
                        {
                        "name": "PARAM2",
                        "type": "number",
                        "value": "",
                        "default": "123"
                        },
                        {
                        "name": "PARAM_LIST",
                        "type": "collection",
                        "value": "",
                        "default": "two",
                        "options": ["one","two","three"]
                        }
                    ]
        }
    return JSON.stringify(returnValue);
}
```

Note: returnValue is where you define the runbook json.

You must define the following parameters:

actionID	The unique ID of the runbook. It must be equal to the name of the file. Spaces are not allowed
author	The author of the runbook
name	The name of the runbook
description	The description of the runbook
component	The list of the components where the runbook can run
summary	An explanation of what the runbook does
commands	A list of commands that will run on the machine
os	The operative system where the runbook can run
rating	The runbook usefulness. It is updated by the users
docLink	Link to the documentation
readonly	If true, the runbook script cannot be modified. Variables and parameters still work
parameters	The list of the parameters (name and type) inserted by the users from the UI

After a new Runbook is added, run the following commands from the <install-path> directory:

1. **docker stop hero-tomcat**
2. **docker start-tomcat**

Training and Prediction

Training and prediction processes run automatically inside the hero-predictor container every 6 hours.
By default, predictions are made for the next 24 hours and requires at minimum 20 mins.
To obtain meaningful predictions, it is recommended to have at least 10 hours of KPI collection.
User has the option to manually trigger predictions for individual machines.

Configuring Security

HERO user authentication is managed by Keycloak.
In Keycloak, each application has its own Realm with different users and authorization settings. HERO authorization settings are stored in a Realm named HERO.
In HERO Realm, the nginx client uses Keycloak to manage user authentication to the remote machines providing a single sign-on solution.
For details about Keycloak, see Keycloak documentation.

The steps to configure the security for your HERO 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 HERO users:

- userid **test**, password **test**, with user role
- userid **admin**, password **admin,** with administrator role

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

Configuring Security manually

To properly customize Keycloak in your environment, run the following steps:

1. In the <install-path> directory, find the .tomcat.env file.
2. In the .tomcat.env file, find the variables:
   - CLIENT_SECRET
   - KEYCLOAK_URL
3. In the KEYCLOAK_URL {"realm": "hero","url": "https://hero.hcltech.com/keycloak/auth","clientId": "nginx"}, replace the hero.hcltech.com part with <IP>:<PORT> of the host machine.
   If the port used in the docker-compose.yml file is the default one (443), only the <IP> value must be added.
4. In the <install-path>\CONFIGURATION\HERO\ui.properties configuration file (see Installing and Configuring), set the variable keycloak to the same value assigned to KEYCLOAK_URL in step 3.
5. After the execution of the docker-compose up --d command, the user test, with password test, is automatically created and authorized to access the HERO dashboard **https://<your_host_machine_address:port>/Dashboard.**
6. To create a user with administrator role, see the steps in Creating a new user below.

Creating a new user

The installation process generates a Keycloak default realm named **HERO** and a default client named **nginx.**
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 HERO 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=**password**
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 -> nginx -> 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 **nginx**. 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

Generating a new secret

For security reasons, you are recommended to generate a new client secret, in place of the default one. To generate a new client secret and customize HERO accordingly, run the following steps:

1. From Keycloak administration console, in the left side navigation bar, select **Clients->nginx.**
2. From the tab **Credentials**, click **Regenerate Secret.**

3. Copy the content of the field **Secret.**
4. From the <install-path> directory, run the commands:
   - **docker-compose down**
   - **docker volume rm <install-path>_hero-home** (to remove the configuration volume)
5. In the <install-path>\CONFIGURATION\HERO directory, find the .tomcat.env file.
6. Paste the content you copied from the field **Secret** into the **CLIENT_SECRET** parameter.
7. In the <install-path>\CONFIGURATION\HERO\ui.properties configuration file, paste the same content into the **clientSecret** parameter.
8. From the <install-path> directory, run the commands **docker-compose up --build**

Customizing SSL certificates

To install your own SSL certificates, where default certificates are already in place, follow these steps:

1. Navigate to the <install-dir>/ro/cert/nginx folder.
2. Replace the default certificate files, hero.key and hero.crt, with your custom SSL certificate and key files.
   Note: Ensure that the file names remain unchanged as hero.key and hero.crt.
3. From the <install-path>/hero directory, run the following command to restart the services and update the installation:
   ```
   docker-compose stop
   docker-compose start
   ```

Customizing Keycloak for AppScan

In the Keycloak administration console, selecting **Client -> nginx**, the parameter **Redirect URIs** has a default value set to "*", which means that the login can be redirected to every URI. To avoid this is identified as an issue by AppScan, modify the **Redirect URIs** parameter by adding only the URIs requested by HERO:

```
HOSTNAME\Dashboard
HOSTNAME\tomcat
HOSTNAME\keycloak
HOSTNAME\elasticsearch
HOSTNAME\kibana
HOSTNAME\prediction
```

where HOSTNAME is the hostname of the machine where HERO is installed.

Appendix

This Appendix contains:

1. A procedure to install Docker and Docker Compose.
2. A manual procedure to install HERO: it guides you to manually execute the steps run by the automatic installation script.
3. Information about using DB2 database instead of Derby.
4. NGINX configuration example.

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

Installing HERO manually

To manually install HERO, run the following procedure:

1. Unzip the HERO Docker build file in the <BUILD_DIR> directory.
2. In the <BUILD_DIR>\CONFIGURATION\HERO directory, locate the dashboard.properties and ui.properties configuration files and customize the following properties with the external hostname and ip address of the workstation where HERO is being installed:
1.
    - **dashboard.properties**
        - IPdashboard
        - elasticsearch_external
        - LaunchInContextUrl
    - **ui.properties**
        - IP
        - kibanaHost
        - keycloak
3. Configure HERO Security as described in Configuring Security.
4. From the <BUILD_DIR> directory, type **docker-compose up -d** to start HERO.
5. Type **https://<your_host_machine_address>/Dashboard** to access HERO dashboard (can be default access port 443, or your custom port).
6. Before adding environments to HERO dashboard, create an OpenSearch Dashboards default index pattern by running the following steps:
    a. Connect to OpenSearch Dashboards console by typing: **<host:port>/OpenSearch Dashboards/**
    b. On the navigation bar, click **Dev Tools**
    c. On the left text area, write the following text: “PUT default”
    d. Click the green arrow
    e. On the navbar, enter **Management**
    f. Click **Index Patterns**
    g. In the index pattern field write default* then, click **Next Step**
    h. Click the **Create Index Pattern** button
    i. Now that you have created the index pattern, verify that OpenSearch Dashboards uses it as a default: on the left list, if the index pattern presents a star, the configuration is complete
    j. If the star is not present, select the index pattern and click the star button at the top right of the page
7. Hero installation has been completed: you can start adding environments.
8. To stop HERO, from the <BUILD_DIR> directory, type **docker-compose stop**

9. To restart HERO, from the <BUILD_DIR> directory, type **docker-compose start**

Using DB2 database instead of Derby

If you want to use DB2 database instead of Derby, run the following steps:

1. Create the database instance
2. Run the following scripts on the database instance:
    - dashboard.sql (to create the default schema)
    - componentActions.sql (default data)
    - componentGlobalAction.sql (default data)
    - scripts.sql
3. Add the JDBC driver jar files in the <BUILD_DIR>\CONFIGURATION\HERO\lib directory
4. In the <BUILD_DIR>\CONFIGURATION\HERO\conf directory, edit the **context.xml** file and add the following text in the <Context> tag:

```
<Resource name="jdbc/console"
     global="jdbc/console"
     auth="Container"
     type="javax.sql.DataSource"
     driverClassName="com.ibm.db2.jcc.DB2Driver"
     url="jdbc:db2://<host-machine-db2>:<port_db2>/console"
     username="<username_db2>"
     password="<password_db2>"/>
```

NGINX configuration example

The following example shows how to configure an NGINX server for providing authentication and using the HTTPS protocol without configuring each node:

```
upstream tomcat {
     server tomcat:8080 fail_timeout=0;
}
upstream elasticsearch {
   server elasticsearch:9200 fail_timeout=0;
}
upstream kibana {
  server kibana:5601 fail_timeout=0;
}
server { # simple reverse-proxy
      listen       443 ssl default_server;
      ssl_certificate     /usr/share/cert/localhost.cert;
      ssl_certificate_key /usr/share/cert/localhost.key;
      auth_basic_user_file /etc/nginx/.htpasswd;
      auth_basic "Access restricted";

      location /Dashboard {
        proxy_pass      http://tomcat;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
      }
      location /ES {
        rewrite /ES/(.*) /$1  break;
        proxy_pass      http://elasticsearch;
      }
      location /kibana {
        proxy_pass      http://kibana;
      }
}
```

About the User's Guide

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

For detailed how-to user scenarios, see the video HERO for Workload Automation.

Getting started with HERO

After the installation has completed successfully, you need to add the servers to be monitored. Run the following steps:

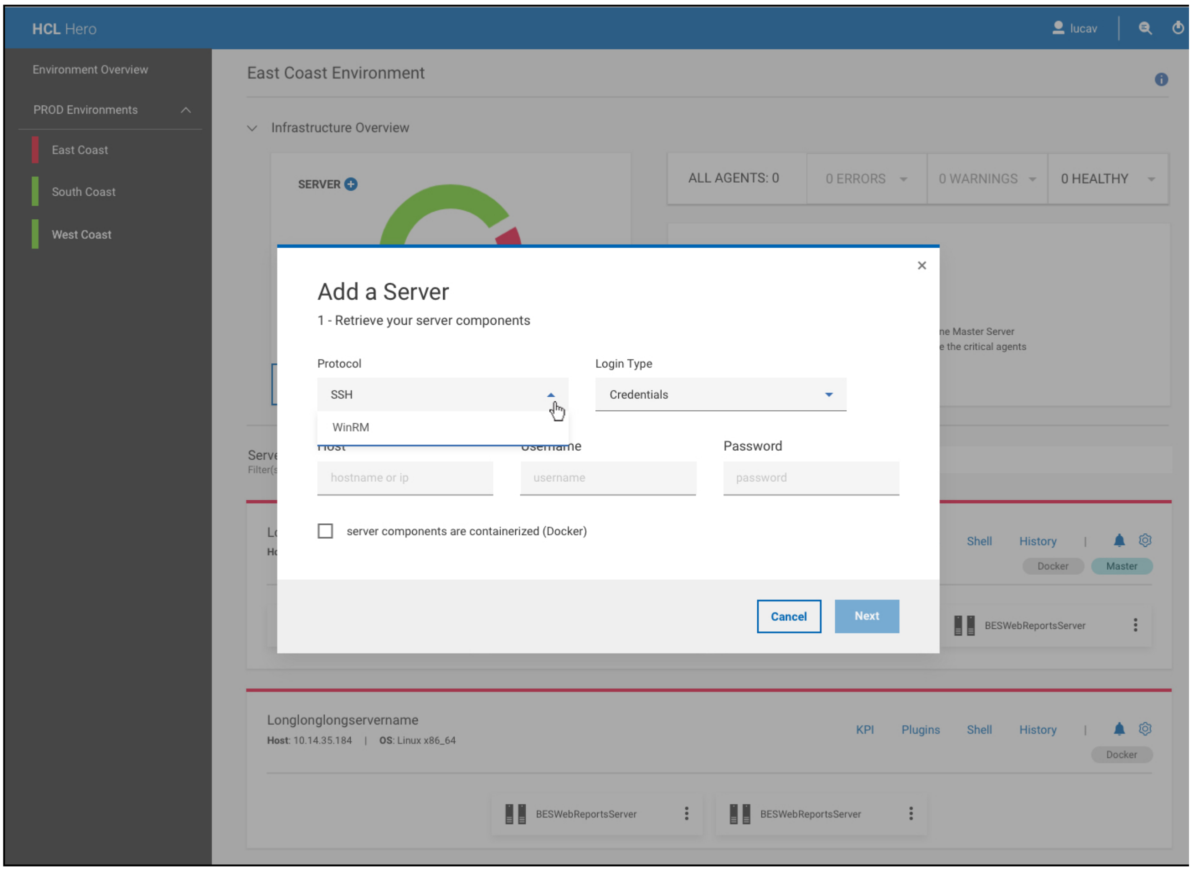
1. Access the HERO dashboard at the following link: **https://<hostname_or_ip>/Dashboard/**.
2. Login with userid **admin**, password **admin.**
3. From the main page, create an environment.
4. Click on the just created environment and add a new server by specifying the required information. For details, see: Adding Servers to your environment.
5. For the discovered server, activate the monitors. Run the following steps::
    a. Click the **monitors** link on the server card.
    b. Activate the selected monitors.
    c. Wait for a few minutes and verify that all the selected monitors become green, red, or yellow.
6. The monitoring scripts are automatically scheduled by HERO on a regular basis.

Adding Servers to your environment

From the Server section of your environment, click **Add Server** to add a new server and retrieve the server components.

You can login to the remote machine in two different ways:

1. **Login with credentials, through SSH or WinRM protocol**
   In this case, enter the server hostname or ip address, the username and password of the owner of the product instance installed on the server.



Add a Server

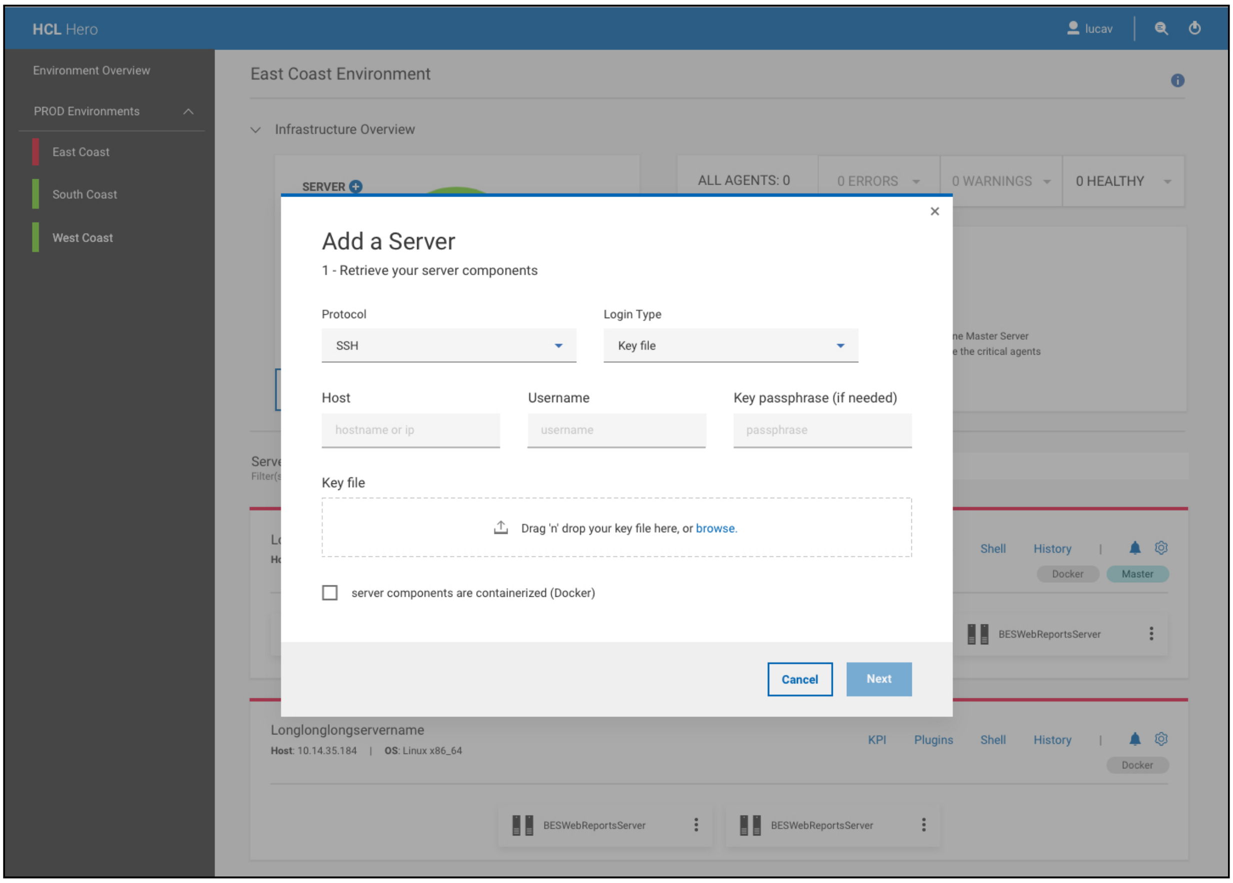
1 - Retrieve your server components

Protocol: SSH

Login Type: Credentials

In this case, enter the server hostname or ip address, the username of the owner of the product instance installed on the server and, optionally, a key passphrase. The passphrase is needed if your private key is passphrase protected. In the dotted area, drag and drop your SSH key file or click **browse** to search the file system. Once you have inserted your SSH key file, you can delete and/or replace it, if needed. The SSH key file can be of any supported format and must contain only the key string.

Your private key is **temporarily** saved by HERO and used only to add the server. After that, your certificate is deleted and replaced by a HERO certificate.



Add a Server

1 - Retrieve your server components

Protocol: SSH

Login Type: Key file

Host: hostname or ip

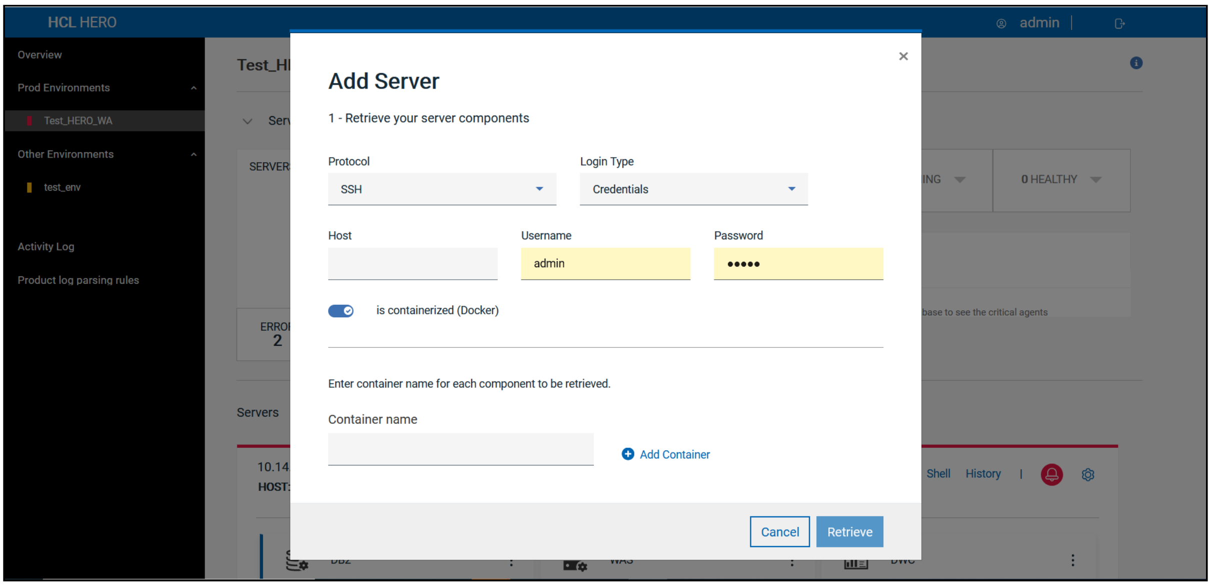
Username: username

Key passphrase (if needed): passphrase

Key file

Drag 'n' drop your key file here, or browse.

□ server components are containerized (Docker)



Add Server

1 - Retrieve your server components

Protocol: SSH

Login Type: Credentials

Host:

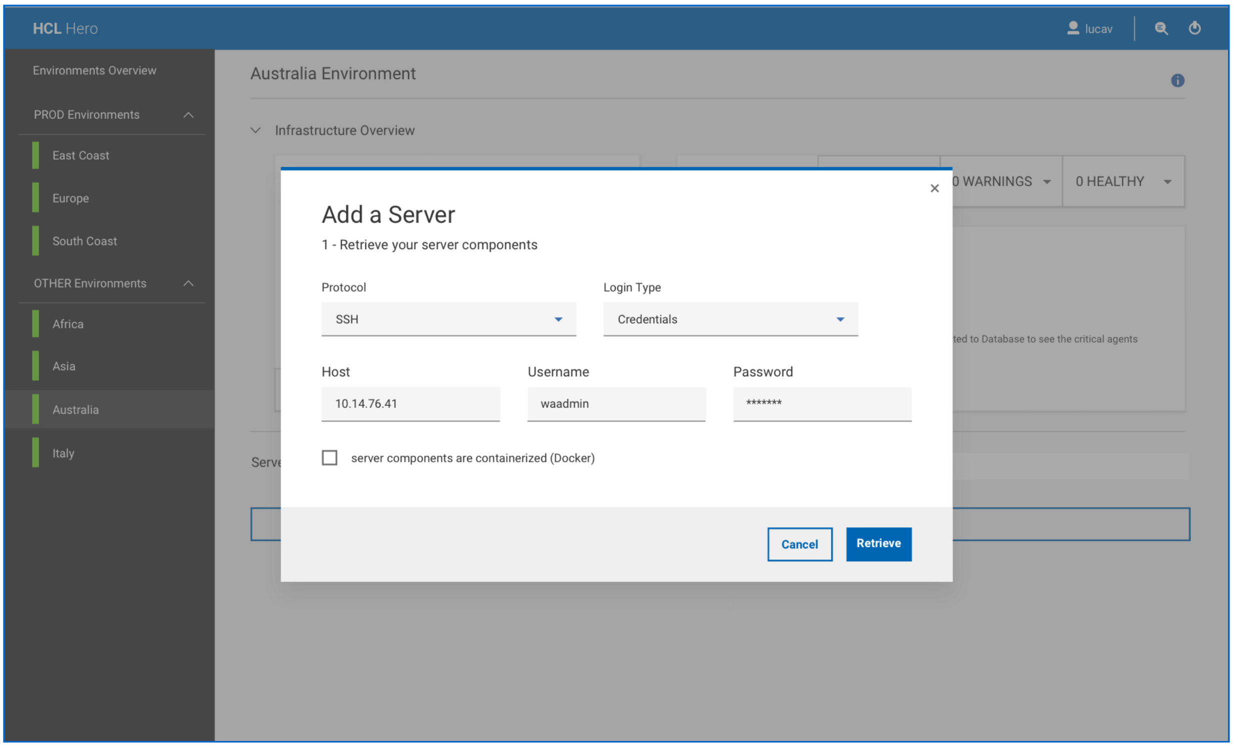
Username: admin

Password: •••••

is containerized (Docker)

Enter container name for each component to be retrieved.

Container name



Add a Server

1 - Retrieve your server components

Protocol: SSH

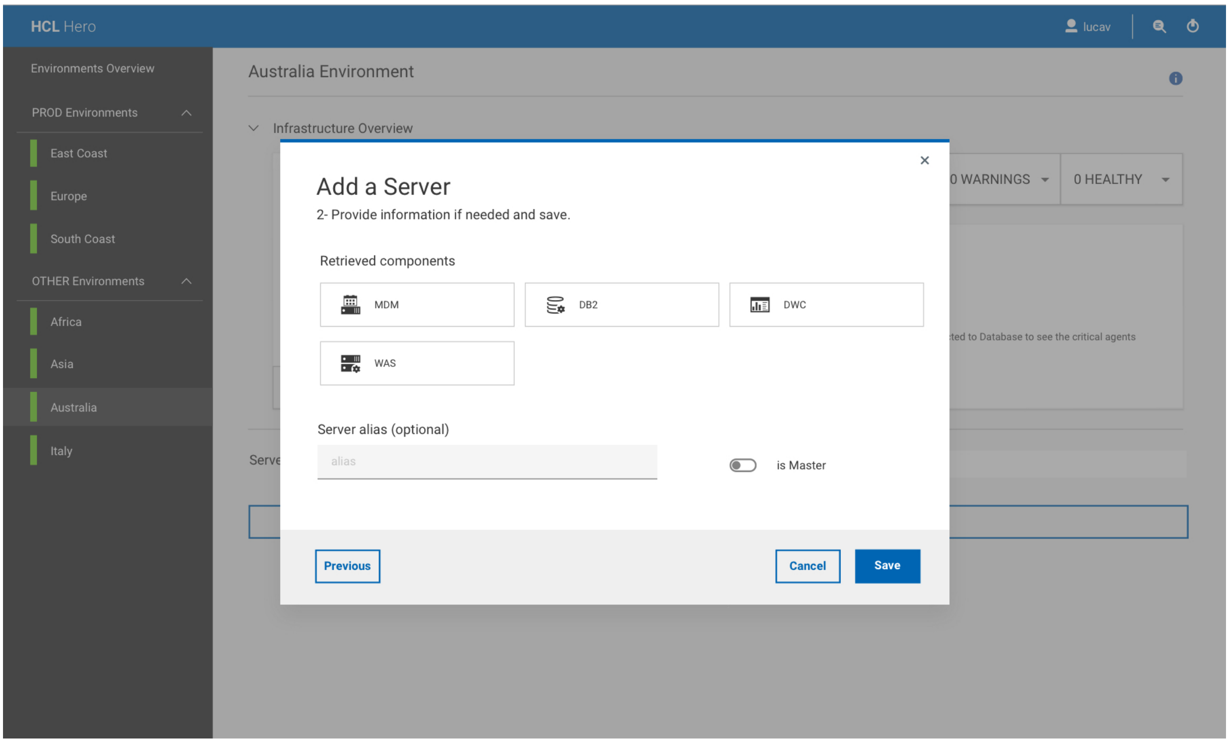
Login Type: Credentials

Host: 10.14.76.41

Username: waadmin

Password: *******

□ server components are containerized (Docker)



Add a Server

2- Provide information if needed and save.

Retrieved components

MDM
DB2
DWC
WAS

Server alias (optional): alias

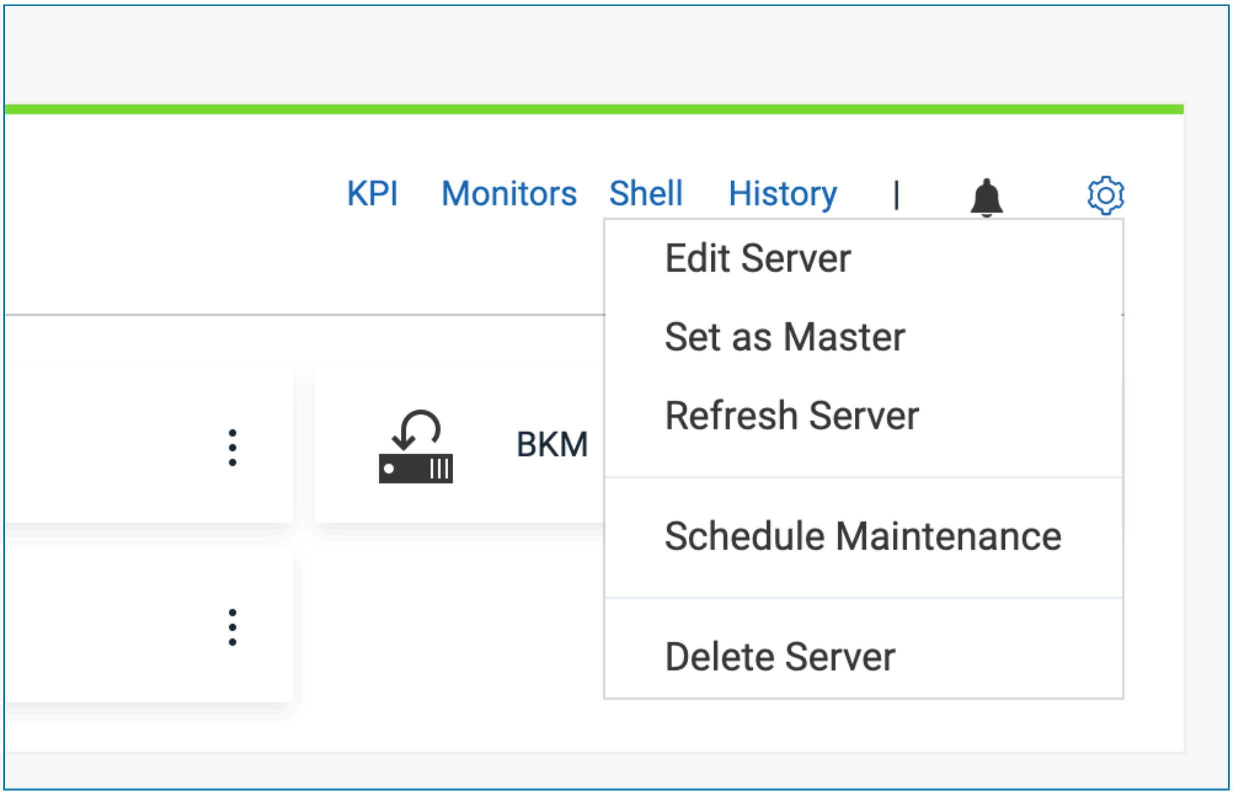
is Master

2. **Login with SSH Key**
3. If the server that you want to add contains containerized components, you must check the related box in the Add Server panel, and provide the container name for each component to be retrieved.

4. After you have successfully logged in to your server, click **Retrieve.** HERO will automatically discover the Workload Automation components deployed on the server.
5. If you flag the server as "Is master", a form will appear to specify the parameters needed to retrieve the list of the critical Agents. Provide the engine hostname, and the username and password of the user enabled to run Workload Automation Rest APIs.
6. Click **Save** to add the server to your environment.

Refreshing Server information

After you have added a server to your environment, you might want to refresh the server information. Run the following steps:

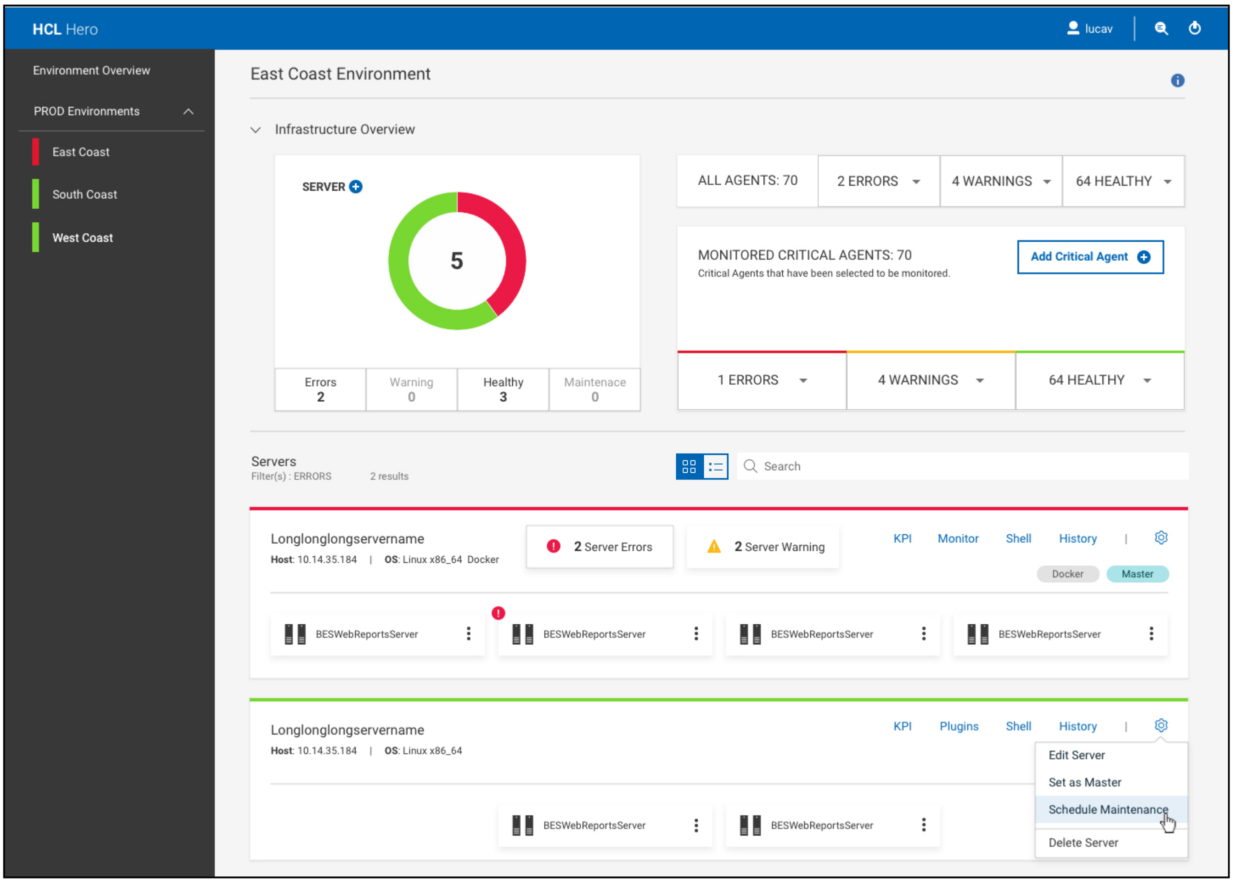


1. From the server drop-down menu, select **Refresh Server**.
2. The retrieve process is run again over the existing server connection.
3. When the retrieve process has been completed, the Save Server panel is shown with updated information from the server.
4. You can edit information, such as: server alias, "is Master" flag, and database information, just like during the usual retrieve process.
5. Save the server with the updated information in your environment.

Scheduling Maintenance

You can schedule server maintenance from the server drop-down menu.

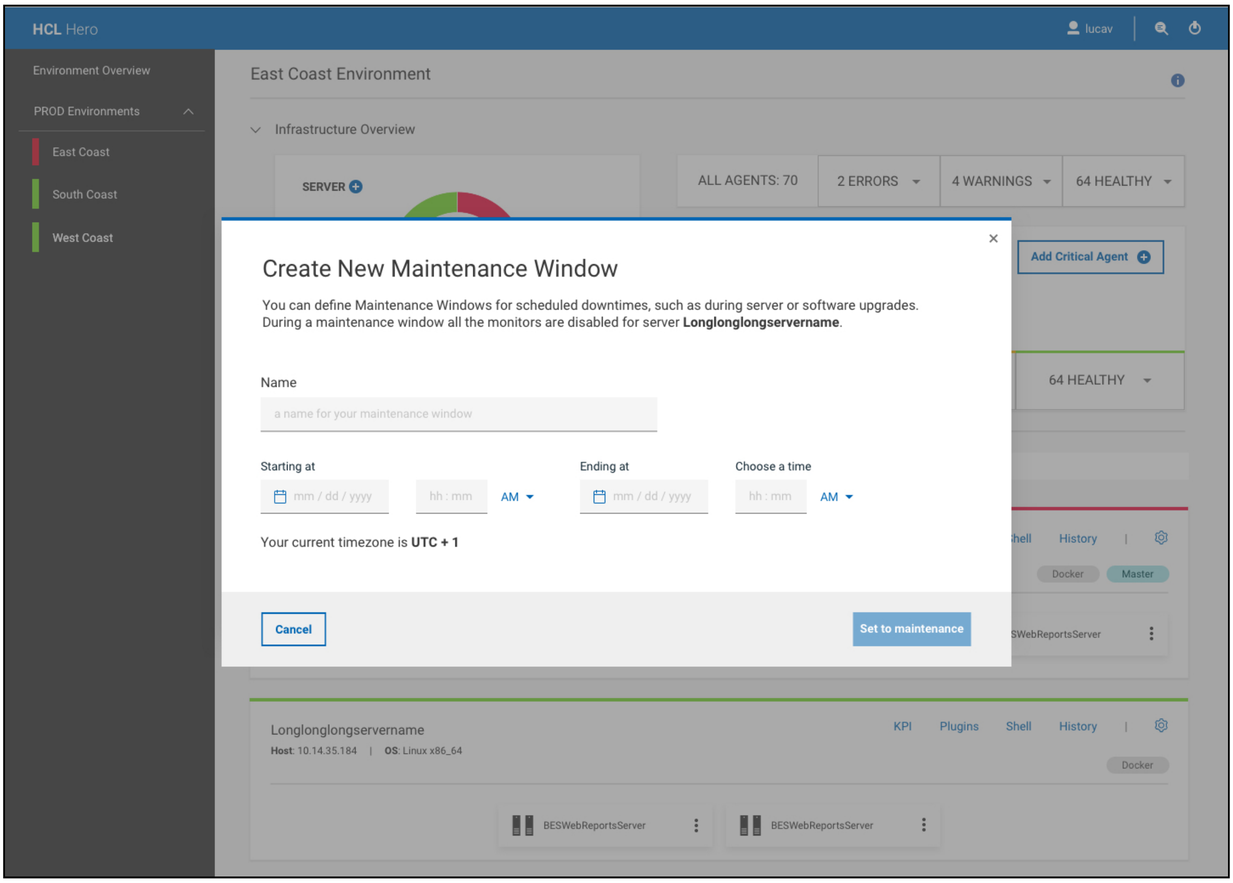
Note: During a maintenance window, all the server monitors are disabled.



In the **Create New Maintenance Window** page, you can set :

- The name of the maintenance window
- The start date and time, the end date and time

Click **Set to maintenance** to enable the maintenance window.



Create New Maintenance Window

You can define Maintenance Windows for scheduled downtimes, such as during server or software upgrades. During a maintenance window all the monitors are disabled for server **Longlonglongservername**.

Name

a name for your maintenance window

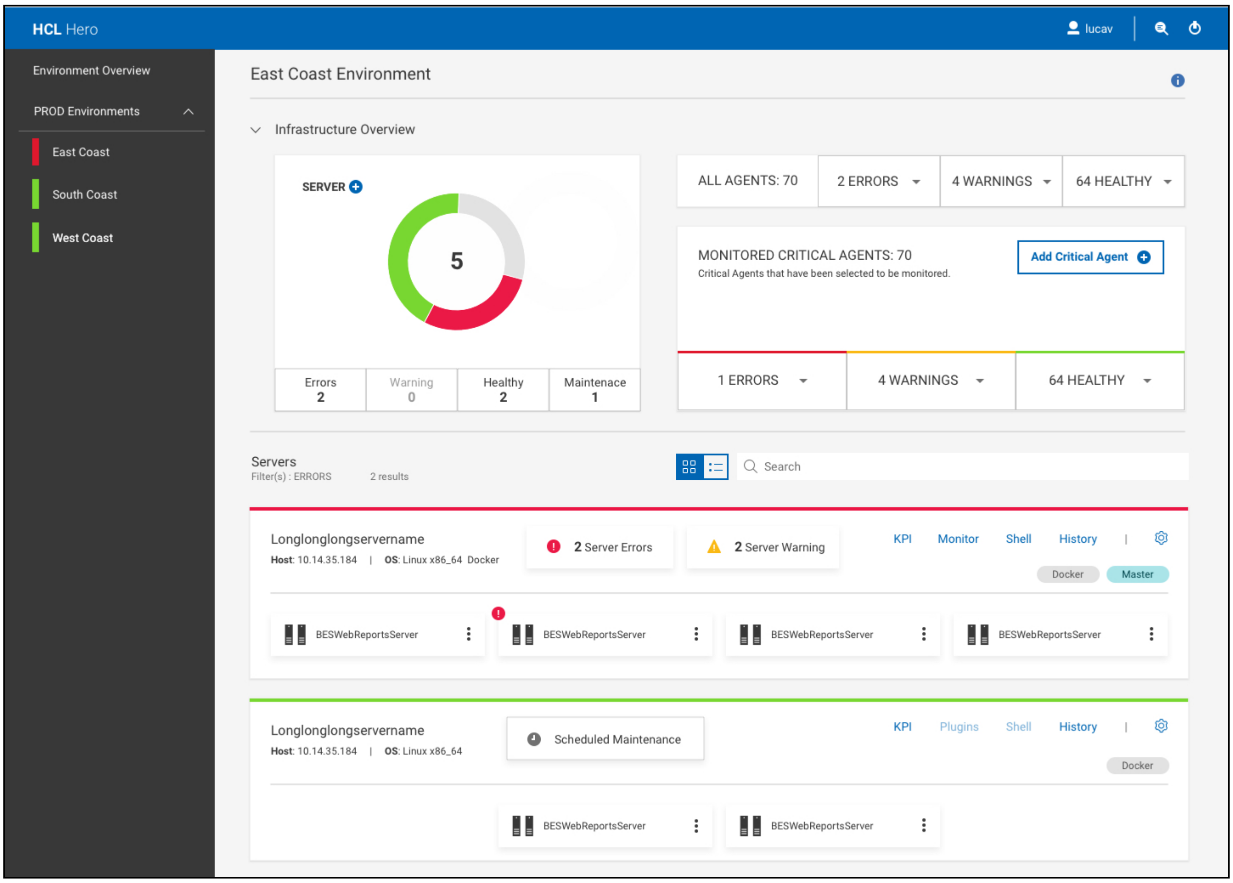
Starting at: mm / dd / yyyy hh : mm AM

Ending at: mm / dd / yyyy

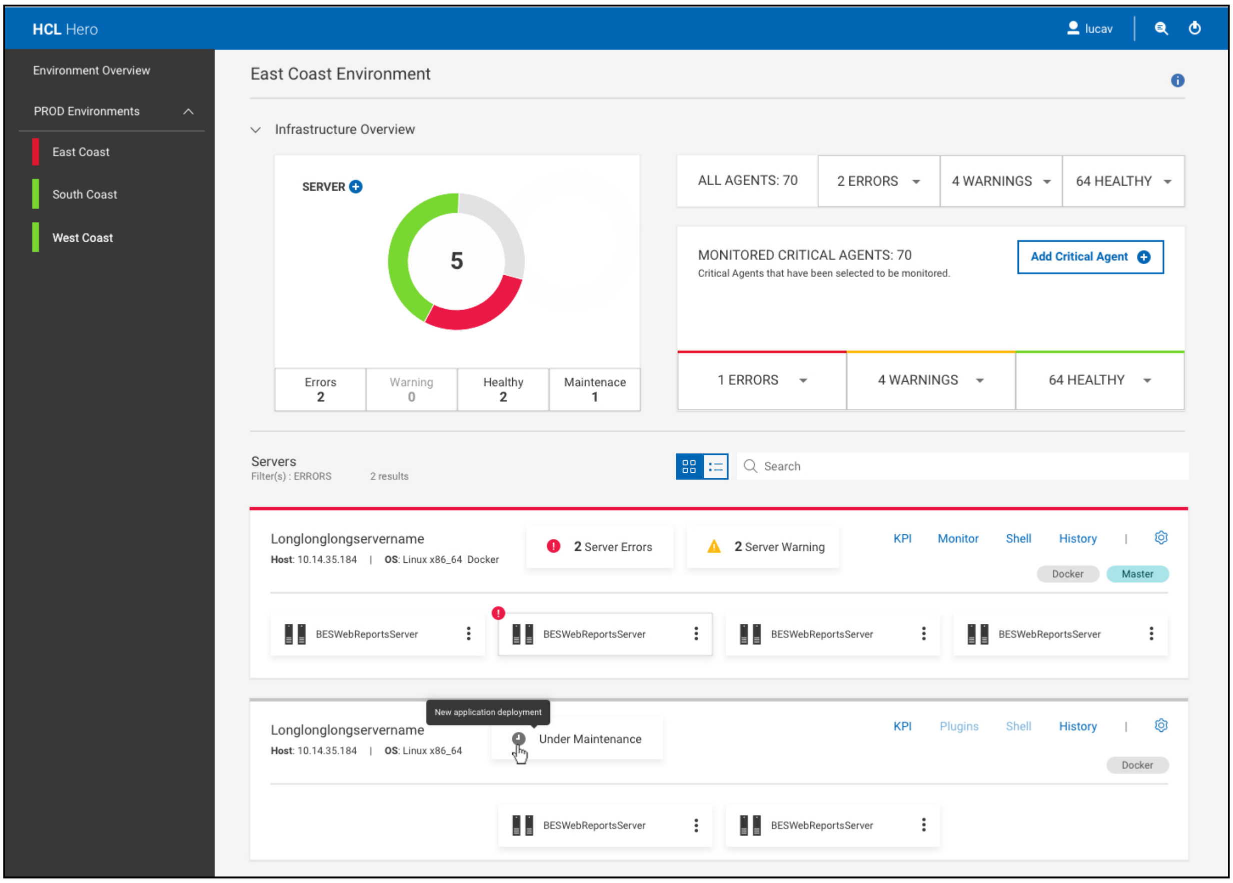
Choose a time: hh : mm AM

Your current timezone is **UTC + 1**

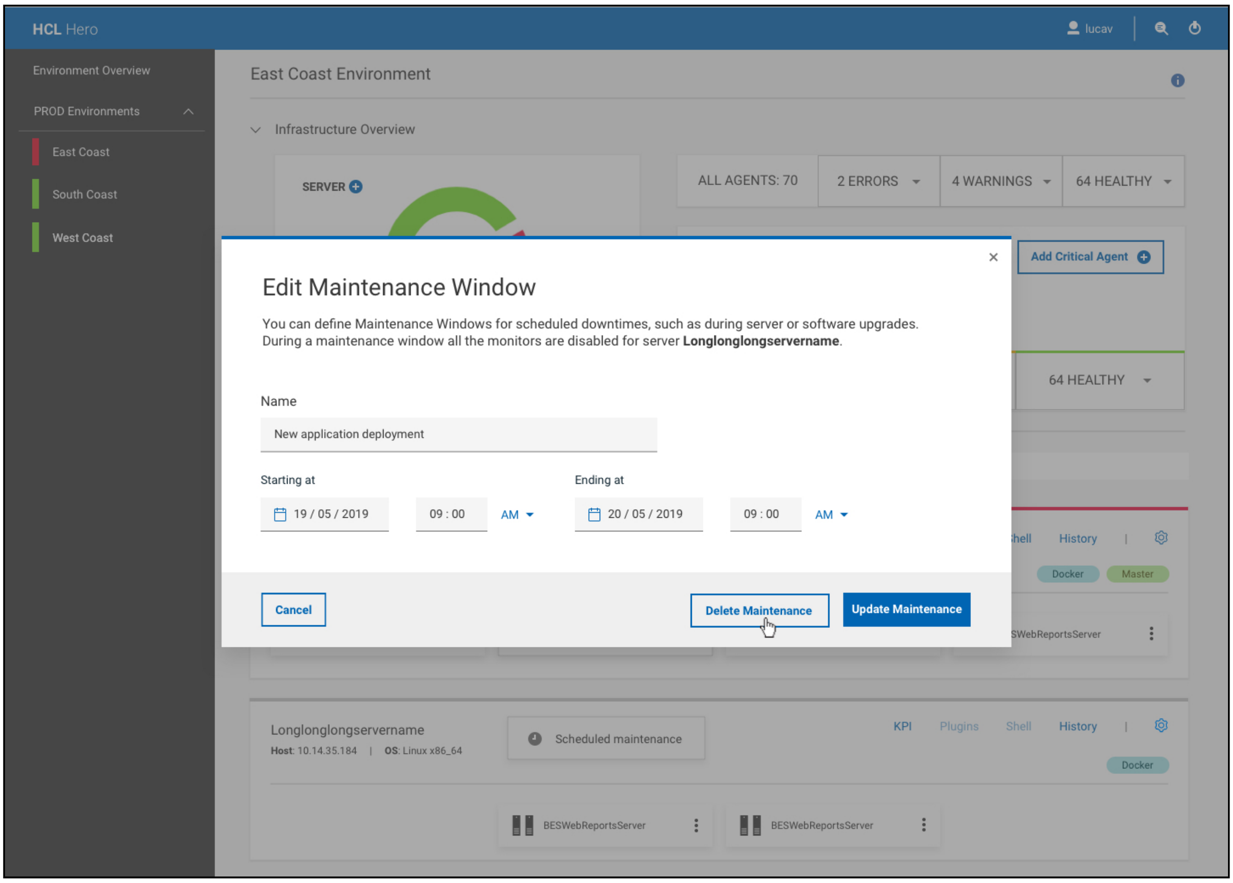
To update the maintenance window, click on the **Scheduled maintenance** card.



When the maintenance is started, the card indicates an **Under maintenance** status. You can still update the maintenance window by clicking the **Under maintenance** card.



You can also delete a scheduled maintenance.



Edit Maintenance Window

You can define Maintenance Windows for scheduled downtimes, such as during server or software upgrades. During a maintenance window all the monitors are disabled for server **Longlonglongservername**.

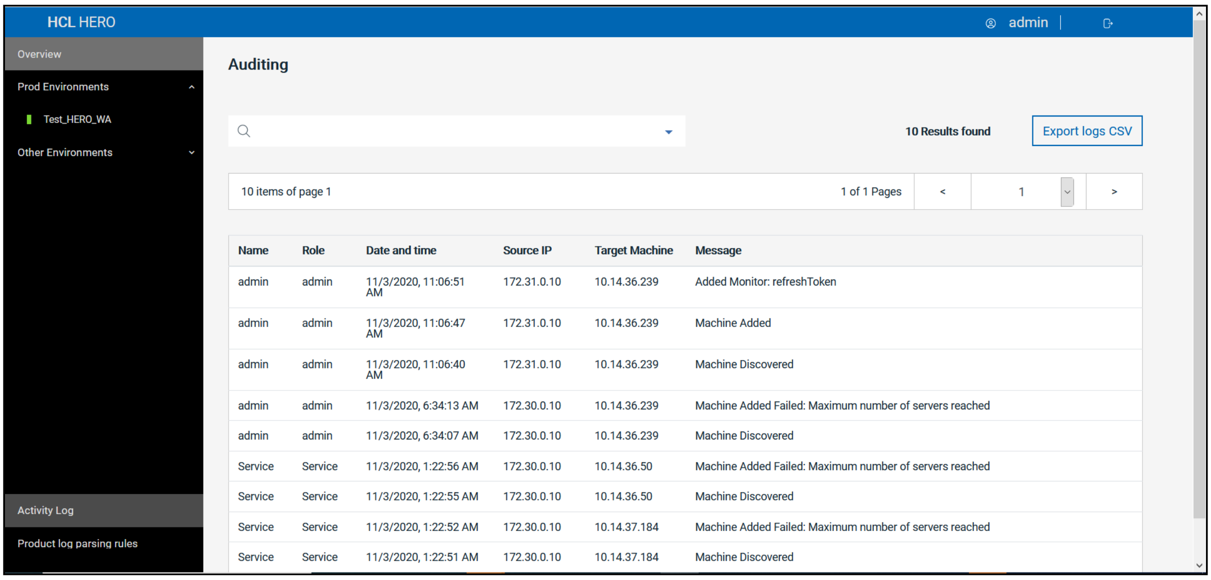
Name: New application deployment

Starting at: 19 / 05 / 2019 09 : 00 AM

Ending at: 20 / 05 / 2019 09 : 00 AM

Auditing

By selecting **Activity Log** on the left hand menu, you can find information about all changes and actions made in HERO.



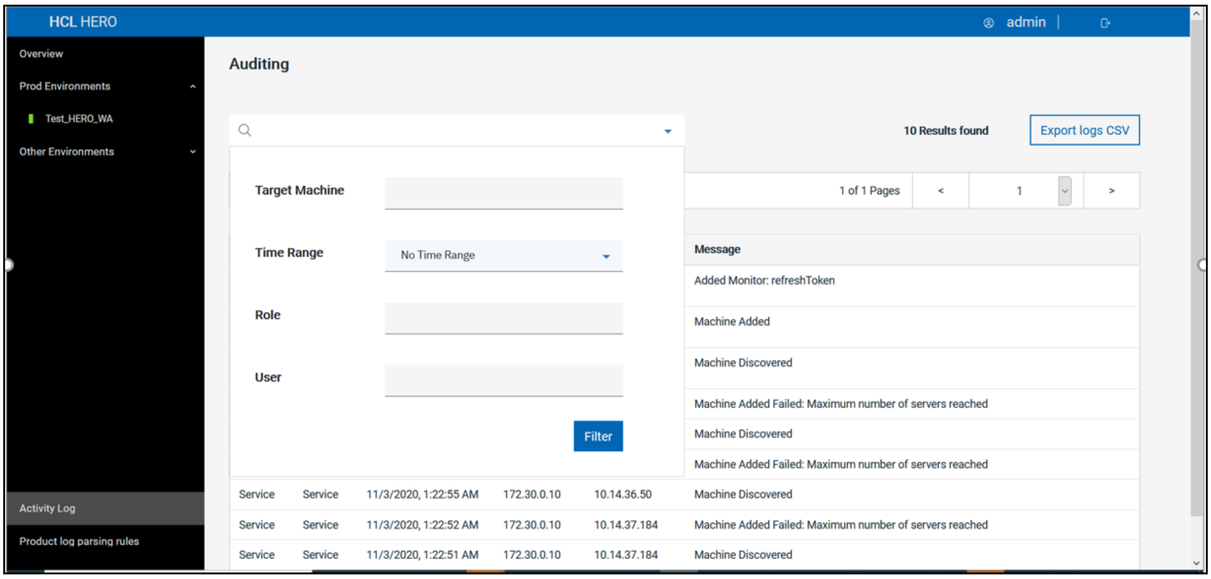
Auditing

10 Results found

10 items of page 1 — 1 of 1 Pages — 1

Name	Role	Date and time	Source IP	Target Machine	Message
admin	admin	11/3/2020, 11:06:51 AM	172.31.0.10	10.14.36.239	Added Monitor: refreshToken
admin	admin	11/3/2020, 11:06:47 AM	172.31.0.10	10.14.36.239	Machine Added
admin	admin	11/3/2020, 11:06:40 AM	172.31.0.10	10.14.36.239	Machine Discovered
admin	admin	11/3/2020, 6:34:13 AM	172.30.0.10	10.14.36.239	Machine Added Failed: Maximum number of servers reached
admin	admin	11/3/2020, 6:34:07 AM	172.30.0.10	10.14.36.239	Machine Discovered
Service	Service	11/3/2020, 1:22:56 AM	172.30.0.10	10.14.36.50	Machine Added Failed: Maximum number of servers reached
Service	Service	11/3/2020, 1:22:55 AM	172.30.0.10	10.14.36.50	Machine Discovered
Service	Service	11/3/2020, 1:22:52 AM	172.30.0.10	10.14.37.184	Machine Added Failed: Maximum number of servers reached
Service	Service	11/3/2020, 1:22:51 AM	172.30.0.10	10.14.37.184	Machine Discovered

The logs can be exported into a **.csv** file.
In the **Search bar** you can refine your search by specifying query parameters.



Target Machine

Time Range: No Time Range

Role

User

					Message
					Added Monitor: refreshToken
					Machine Added
					Machine Discovered
					Machine Added Failed: Maximum number of servers reached
					Machine Discovered
					Machine Added Failed: Maximum number of servers reached
Service	Service	11/3/2020, 1:22:55 AM	172.30.0.10	10.14.36.50	Machine Discovered
Service	Service	11/3/2020, 1:22:52 AM	172.30.0.10	10.14.37.184	Machine Added Failed: Maximum number of servers reached
Service	Service	11/3/2020, 1:22:51 AM	172.30.0.10	10.14.37.184	Machine Discovered

Log Parsing and Alert Setting

With HERO, Workload Automation log files can be parsed for an easier and faster analysis. You can activate this feature on all WA servers, **except AIX servers**. See Enabling Log Parsing.

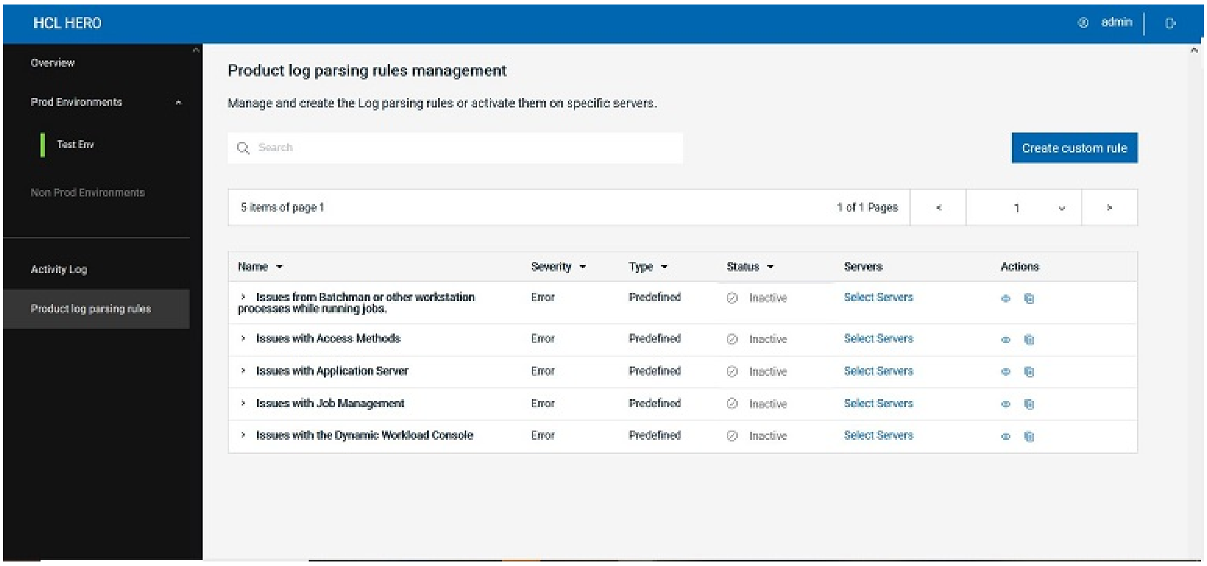
From the **Product log parsing rules management** interface, you as an **Administrator** can create, edit, and delete log parsing rules, and activate them on selected servers.

Alerts can be automatically set up according to your needs.

Log parsing rules can also be activated/deactivated at server level, in the server card. See Activating rules from the server card.

When an issue is detected into a log file, the bell on the server card turns into color (blue, yellow, or red) depending on the issue severity (informational, warning, or error). Click the bell to get information about the issue. See Monitoring Log Parsing rules.

On the HERO main page sidebar, select **Product log parsing rules**.



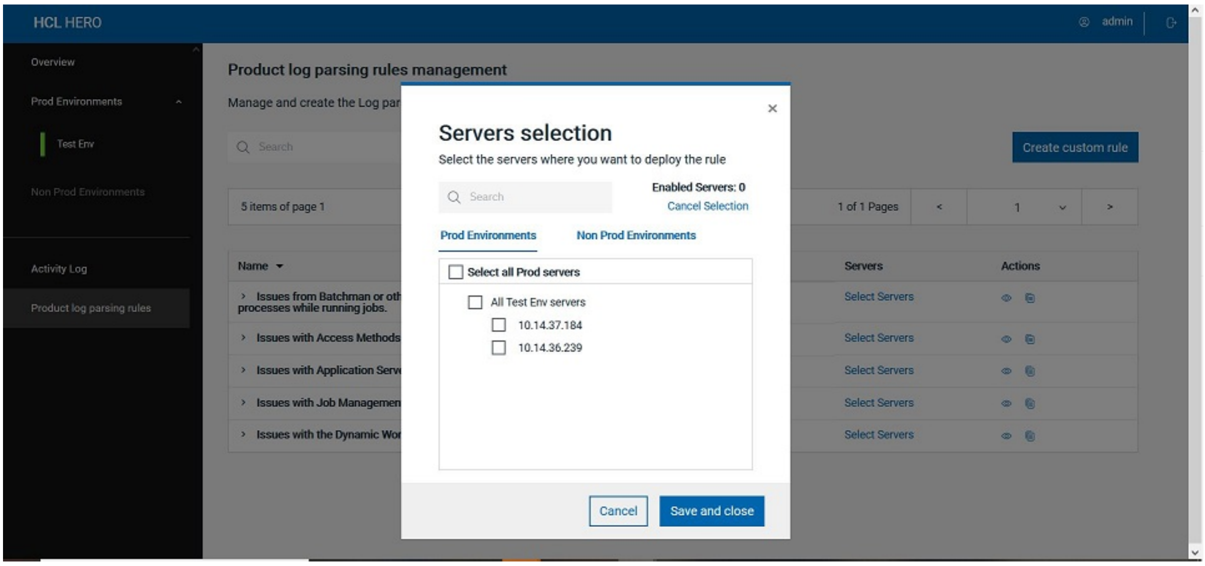
Product log parsing rules management

Manage and create the Log parsing rules or activate them on specific servers.

5 items of page 1 | 1 of 1 Pages | 1

Name	Severity	Type	Status	Servers	Actions
Issues from Batchman or other workstation processes while running jobs.	Error	Predefined	Inactive	Select Servers	
Issues with Access Methods	Error	Predefined	Inactive	Select Servers	
Issues with Application Server	Error	Predefined	Inactive	Select Servers	
Issues with Job Management	Error	Predefined	Inactive	Select Servers	
Issues with the Dynamic Workload Console	Error	Predefined	Inactive	Select Servers	

Here you can find **predefined log parsing rules** provided by HCL, that cover the most common issues with Workload Automation. Predefined rules can be viewed, or duplicated to obtain custom rules. Custom rules can be viewed/edited, duplicated, or deleted. With custom rules you can monitor any log file in your file system, not only WA log files.
You can activate each rule on one or multiple servers by clicking **Select Servers**. You can also select all servers in PROD or OTHER type environments with a single check mark.



Servers selection

Select the servers where you want to deploy the rule

Enabled Servers: 0

Cancel Selection

□ Select all Prod servers
- □ All Test Env servers
  - □ 10.14.37.184
  - □ 10.14.36.239

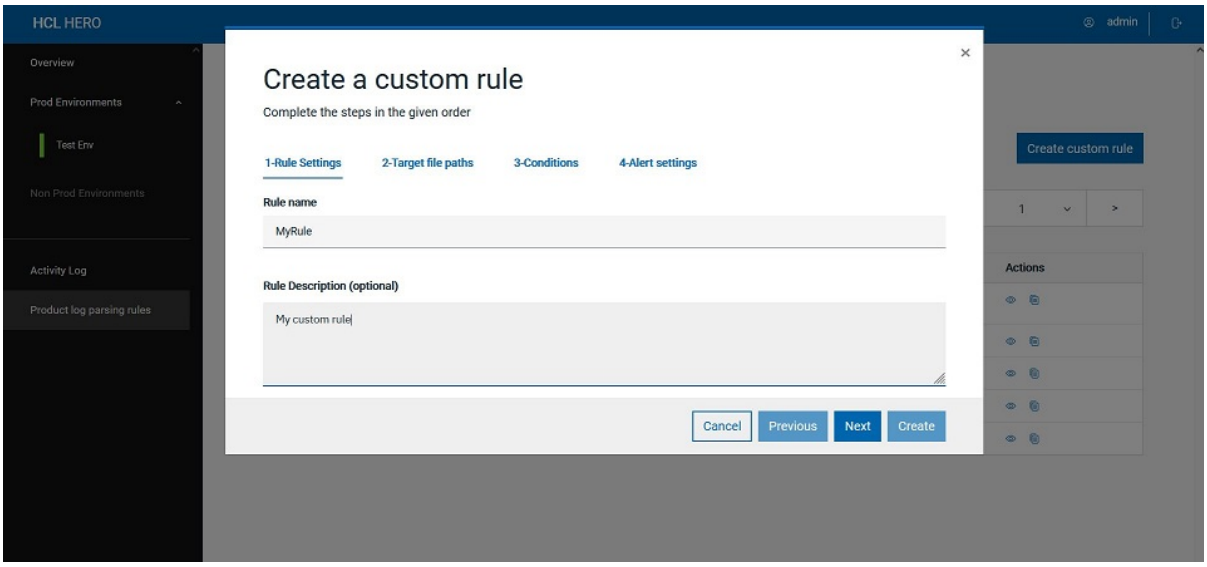
When you activate a rule on a server (master or backup master), the rule applies to all the server components.
For each rule, the following information is displayed:

- Rule name and description
- Severity - can be Informational, Warning, Error

- Type - can be Predefined or Custom
- Author - name of the user who created the rule (HCL for predefined rules)
- Activation - indicates if the rule has been activated or not
- Servers - for activated rules, it indicates the number of servers on which the rule has been activated.

Creating custom rules

On the Log Parsing main page, click the button **Create custom rule** and complete the steps in the given order.



Create a custom rule

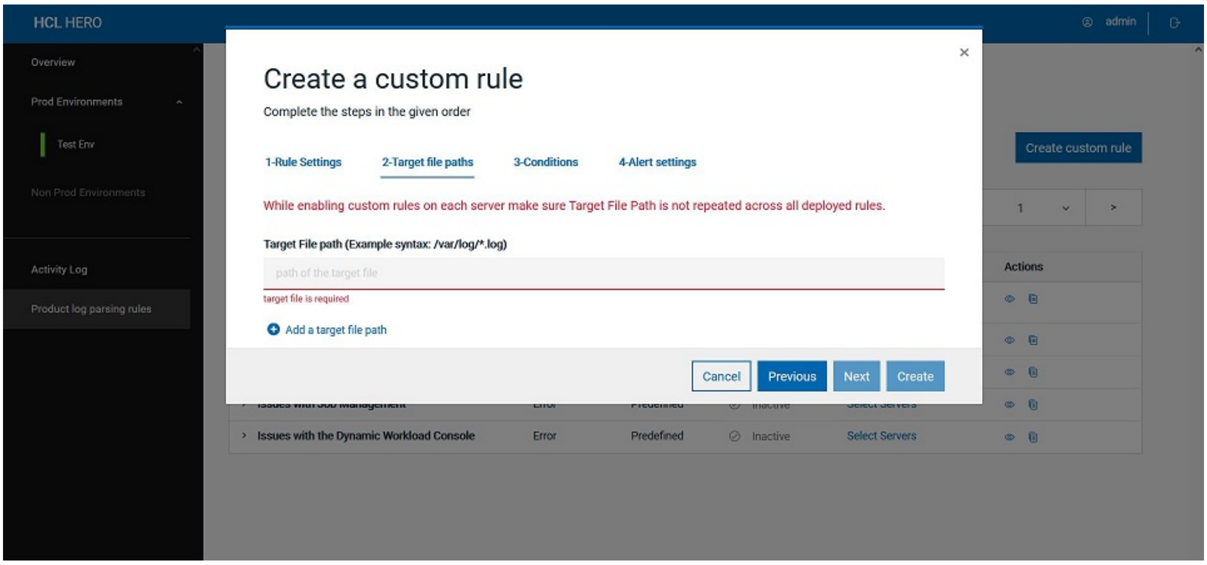
Complete the steps in the given order

Rule name: MyRule

Rule Description (optional): My custom rule

In the **Rule settings tab**, specify:

- Rule name
- Rule description (optional)



Create a custom rule

Complete the steps in the given order

While enabling custom rules on each server make sure Target File Path is not repeated across all deployed rules.

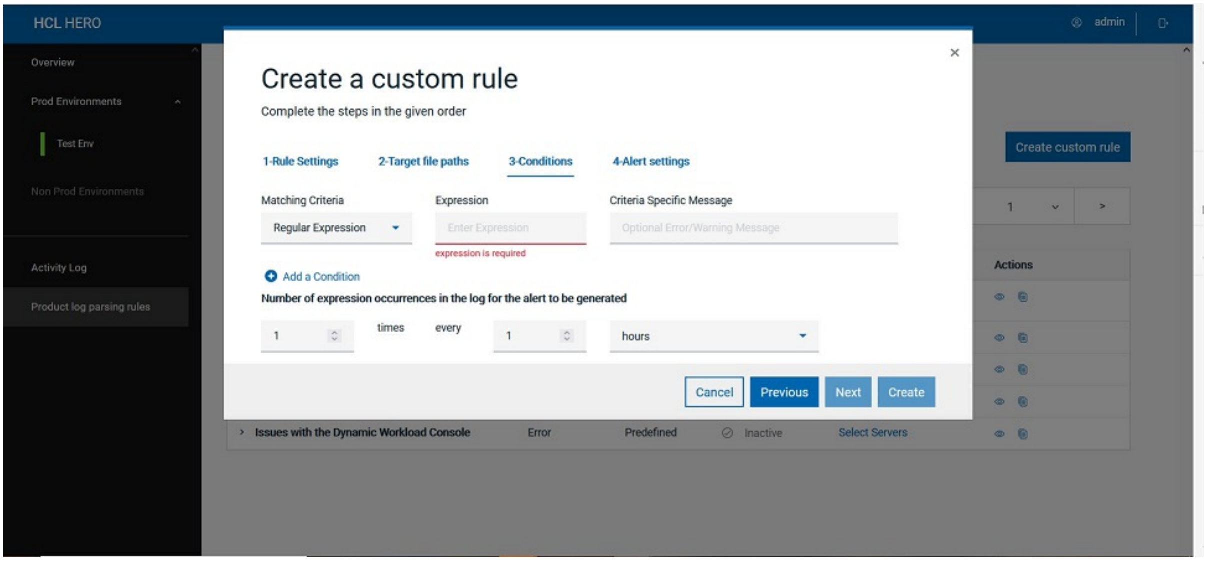
Target File path (Example syntax: /var/log/*.log)

path of the target file

target file is required

Issues with the Dynamic Workload Console	Error	Predefined	Inactive	Select Servers

In the **Target file paths** tab, specify the file path (one or more) for the log file(s) that you want to monitor in the file system.



Create a custom rule

Complete the steps in the given order

1-Rule Settings | 2-Target file paths | 3-Conditions | 4-Alert settings

Matching Criteria: Regular Expression

Expression: Enter Expression

expression is required

Criteria Specific Message: Optional Error/Warning Message

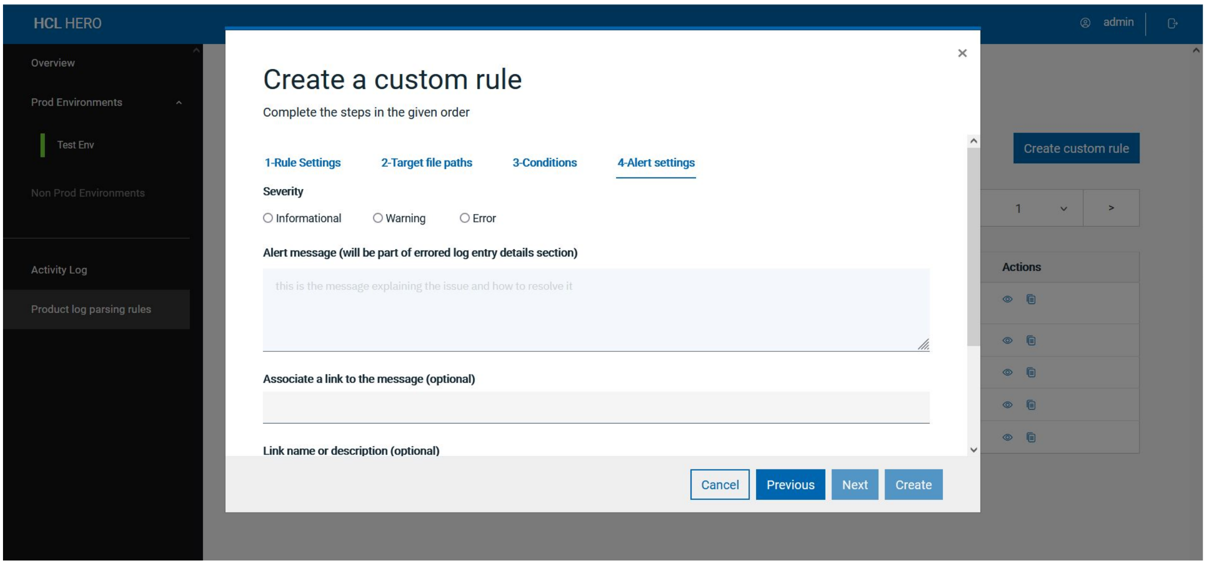
Add a Condition

Number of expression occurrences in the log for the alert to be generated

1 times every 1 hours

Issues with the Dynamic Workload Console | Error | Predefined | Inactive | Select Servers

In the **Conditions** tab, specify one or more expression(s) that you want to match inside log files, the related matching criteria(s) (if regular or wildcard based), and the criteria specific error or warning message (optional).
Set also the number of expression occurrences in the log for the alert to be generated.



Create a custom rule

Complete the steps in the given order

1-Rule Settings | 2-Target file paths | 3-Conditions | 4-Alert settings

Severity

○ Informational ○ Warning ○ Error

Alert message (will be part of errored log entry details section)

this is the message explaining the issue and how to resolve it

Associate a link to the message (optional)

Link name or description (optional)

In the **Alert settings** tab, specify information needed to generate the alert in HERO UI and to notify the alert via email:

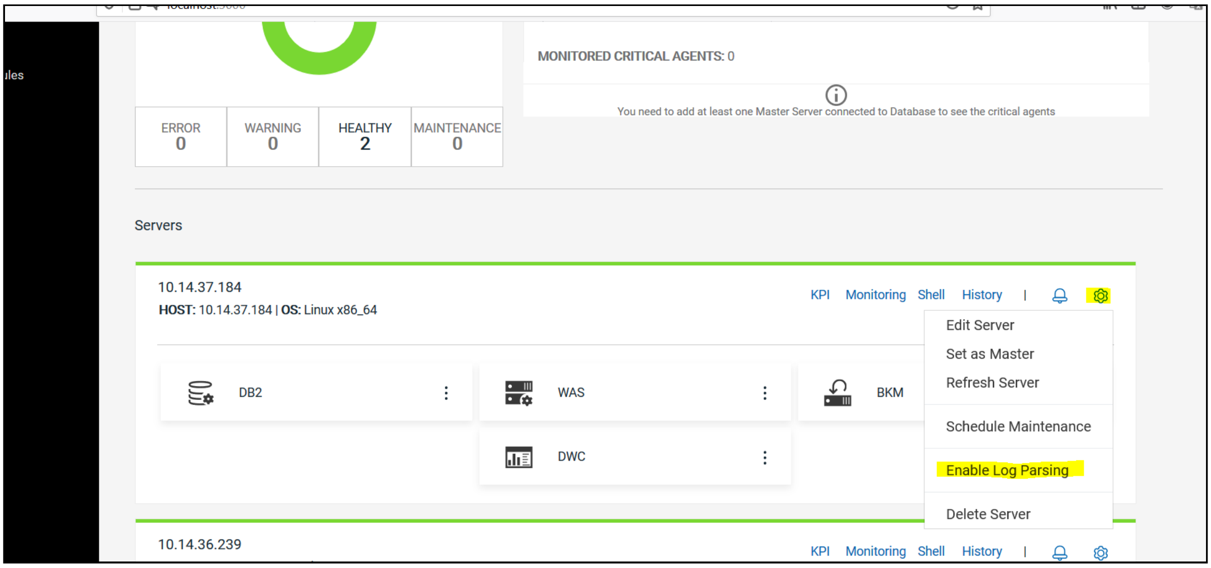
- Alert severity - can be Informational, Warning or Error
- Alert message - to describe the issue and how to solve it (optional)
- Link - where users can find more info about the issue (optional)

- Link description (optional)
- Tag - to insert one or more tags for the rule. Useful also to categorize the issue when generating a ticket (optional)

Enabling Log Parsing

You can enable Log Parsing on all WA servers, except AIX servers, in different ways:

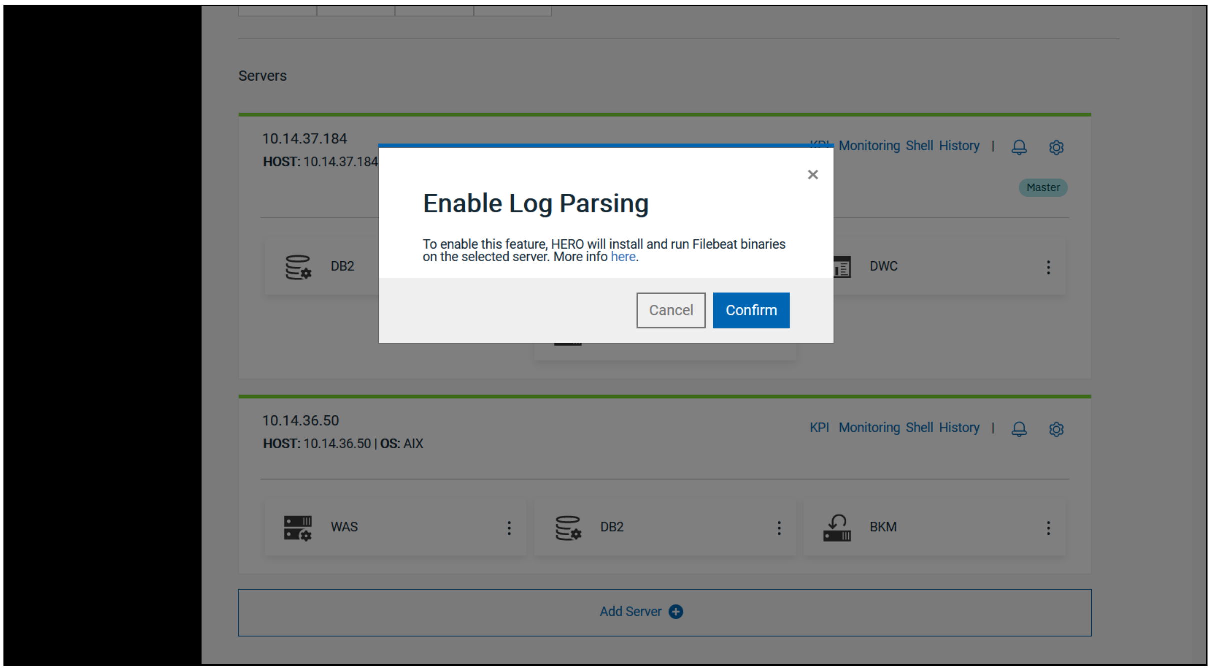
- For a WA server that is already monitored by HERO, click on the gear icon on the server card and select **Enable Log Parsing** from the menu. You can also disable Log Parsing from here, if it is already enabled.



MONITORED CRITICAL AGENTS: 0

You need to add at least one Master Server connected to Database to see the critical agents

HOST: 10.14.37.184 | OS: Linux x86_64

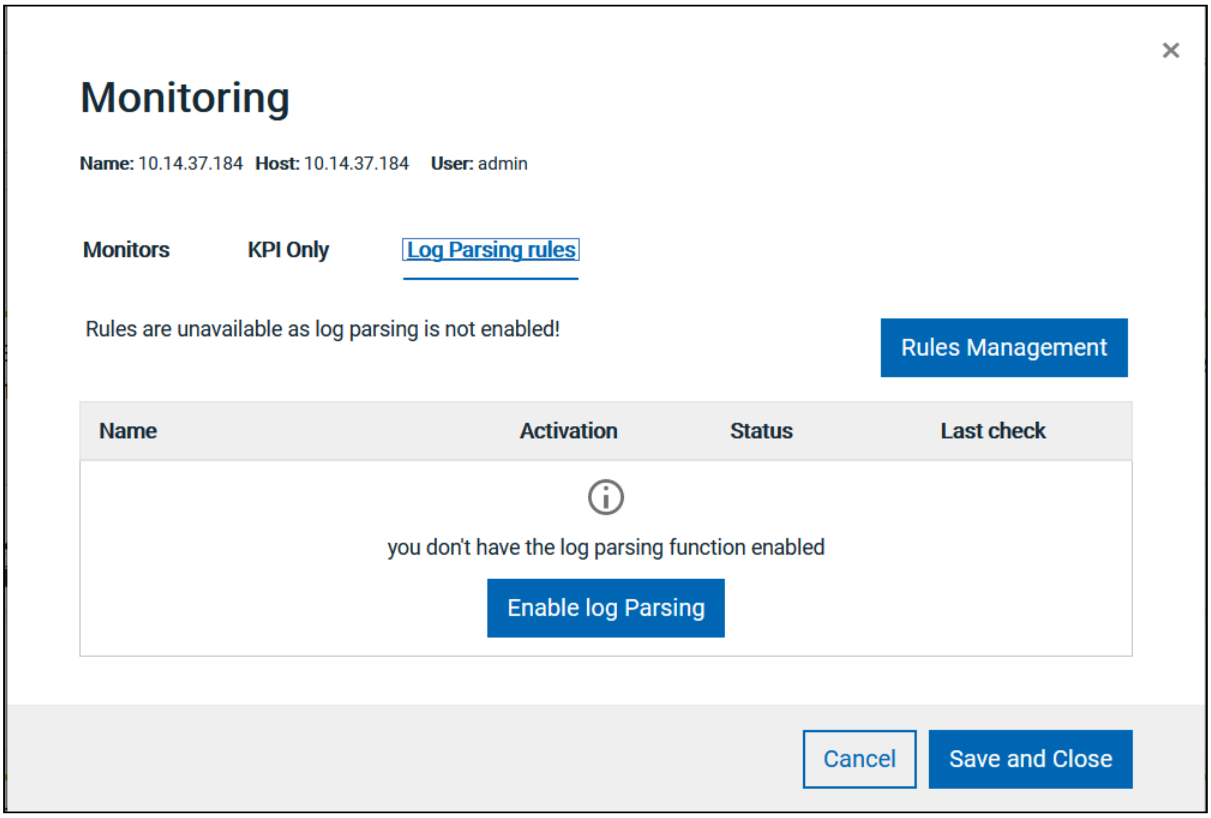


Enable Log Parsing

To enable this feature, HERO will install and run Filebeat binaries on the selected server. More info here.

HOST: 10.14.36.50 | OS: AIX

- Alternatively, from the server card select **Monitoring > Log Parsing rules** and click **Enable Log Parsing.**



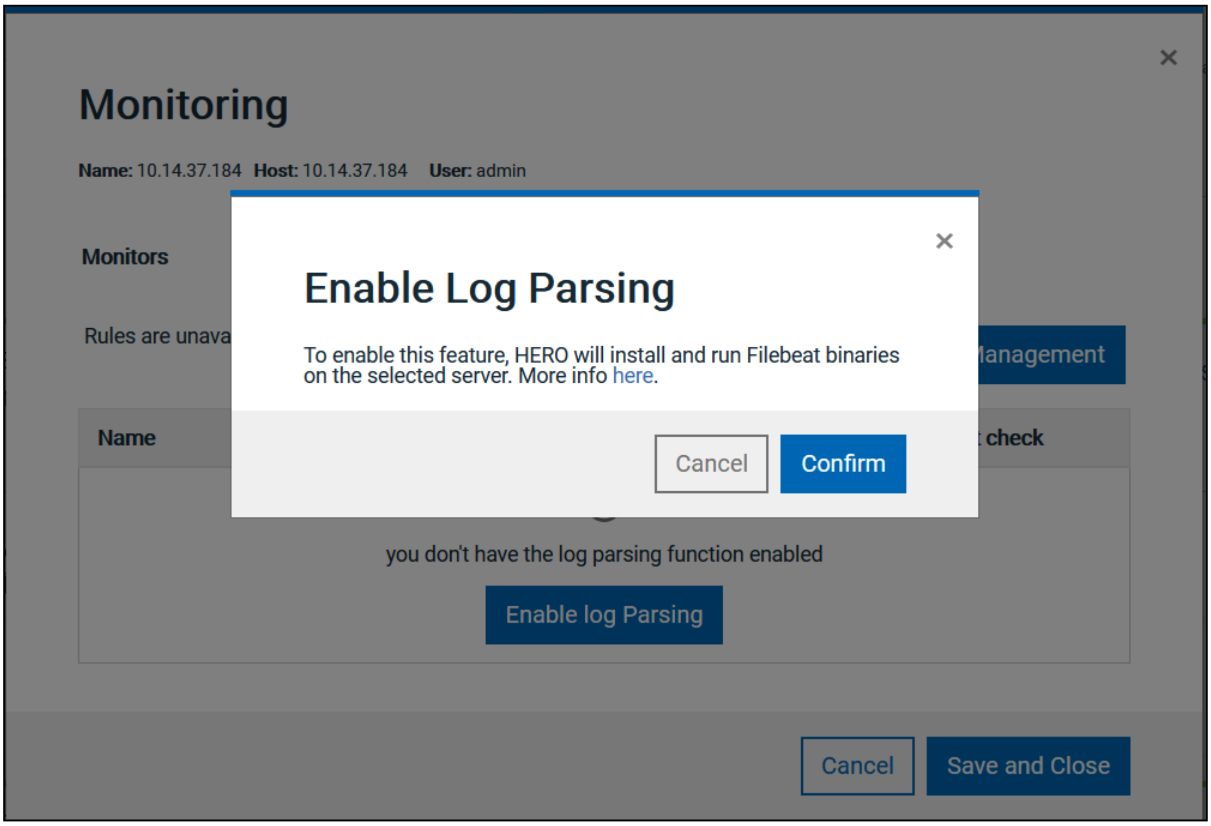
Monitoring

Name: 10.14.37.184 **Host:** 10.14.37.184 **User:** admin

Rules are unavailable as log parsing is not enabled!

Name	Activation	Status	Last check

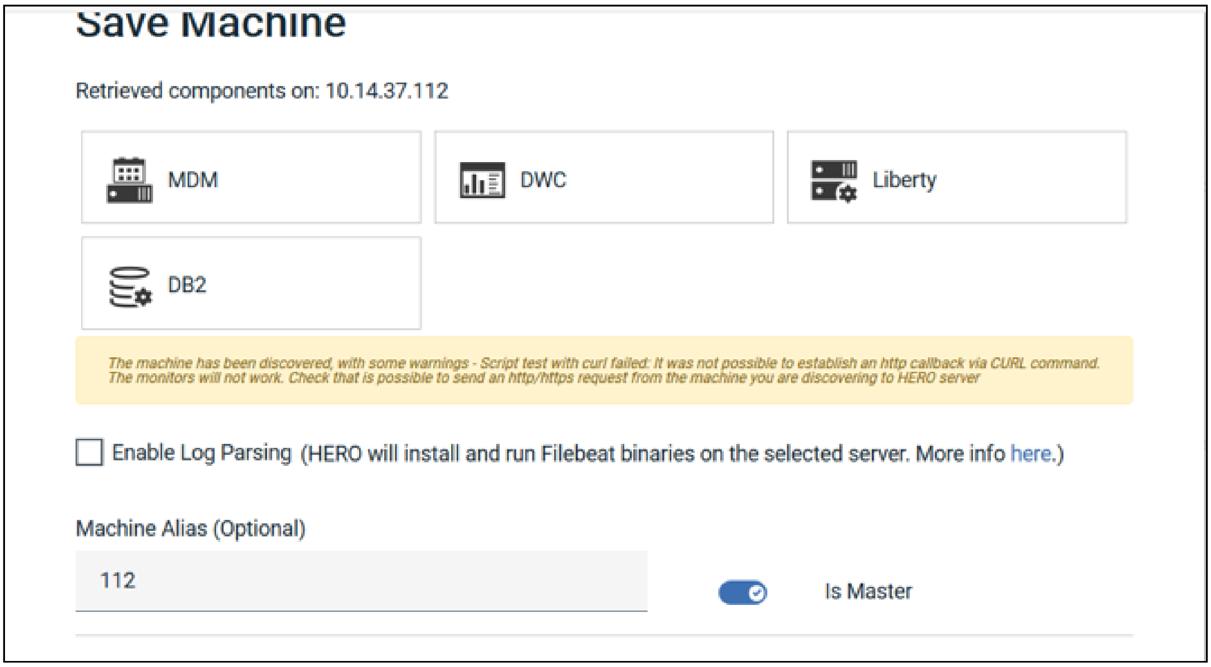
you don't have the log parsing function enabled



Enable Log Parsing

To enable this feature, HERO will install and run Filebeat binaries on the selected server. More info here.

- If you are **adding a new server** to your environment, you can automatically enable this feature by flagging the **Enable Log Parsing** checkbox on the **Save Machine** panel.



Save Machine

Retrieved components on: 10.14.37.112

The machine has been discovered, with some warnings - Script test with curl failed: It was not possible to establish an http callback via CURL command. The monitors will not work. Check that is possible to send an http/https request from the machine you are discovering to HERO server

□ Enable Log Parsing (HERO will install and run Filebeat binaries on the selected server. More info here.)

Machine Alias (Optional)

112

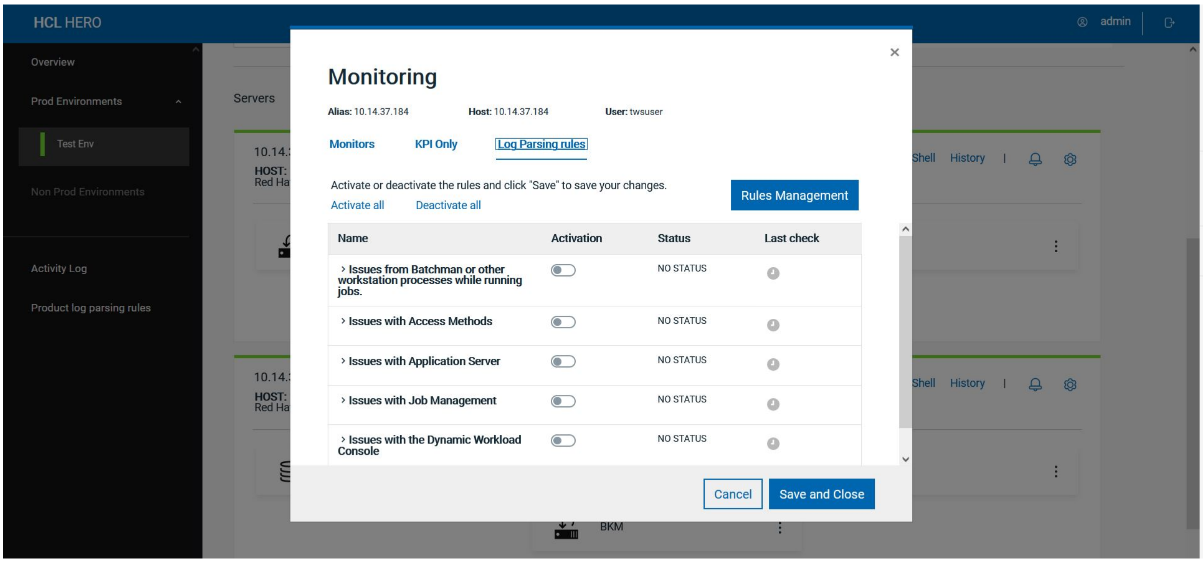
Is Master

In any case, to enable Log Parsing, HERO will install and run **Filebeat** binaries on the selected server.

Note: Filebeat is an open source data shipper for forwarding and centralizing log file data.

Activating rules from the server card

From the server card, under **Monitoring > Log parsing rules**, you can find all the available rules and verify which rules have been activated for your server.



Monitoring

Alias: 10.14.37.184 **Host:** 10.14.37.184 **User:** twsuser

Activate or deactivate the rules and click "Save" to save your changes.

Name	Activation	Status	Last check
› Issues from Batchman or other workstation processes while running jobs.	○	NO STATUS	
› Issues with Access Methods	○	NO STATUS	
› Issues with Application Server	○	NO STATUS	
› Issues with Job Management	○	NO STATUS	
› Issues with the Dynamic Workload Console	○	NO STATUS	

You can activate/deactivate rules from here.

Click the **Rules Management** button to return to the Log Parsing main page.

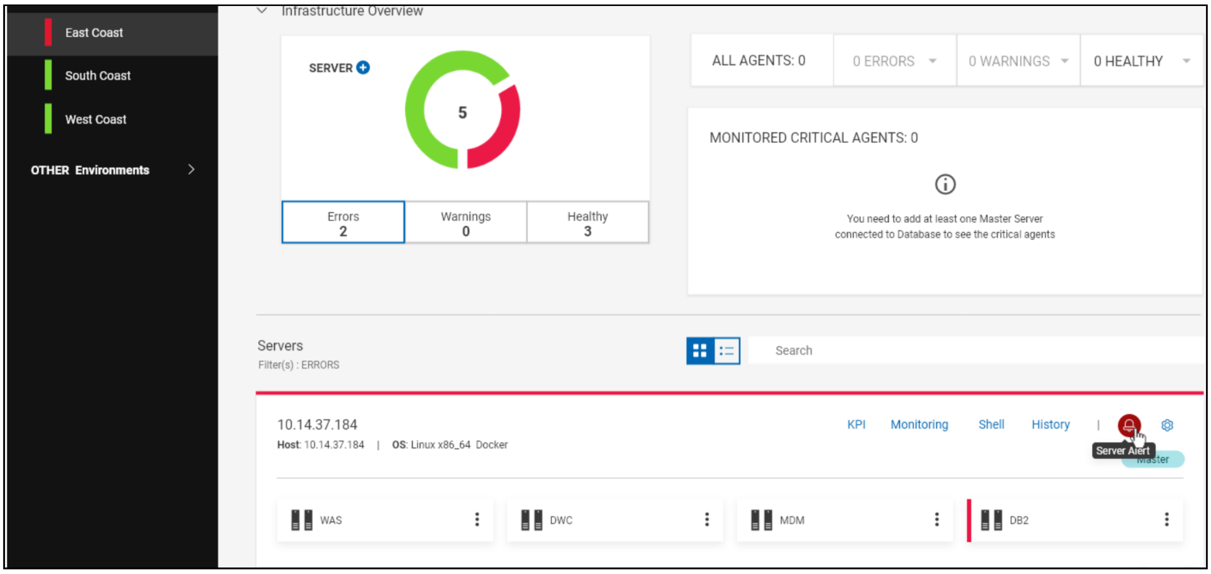
Monitoring log parsing rules

From the server card, under **Monitoring > Log Parsing rules**, you can view the current status of the active Log Parsing rules.

Depending on the alert settings, the **status** of each rule for the server can be:

- **Healthy**, if no issue has been detected by the rule
- **Info**, marked in blue
- **Warning**, marked in yellow
- **Error**, marked in red

Log Parsing alerts are managed by HERO as server alerts: when an issue is detected in a log file, the bell on the server card turns into color (blue, yellow, or red) depending on the issue severity (informational, warning, or error).



You need to add at least one Master Server connected to Database to see the critical agents

Click the bell to get information about the alert.

In the **Server Alerts** panel, you can easily identify Log Parsing alerts because they are always assigned to the **Server** component and the description field contains information about the referenced log entry.

On the same panel, HERO might suggest runbooks to solve the issue.

About the Troubleshooting Guide

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

Collecting logs and activating traces

Troubleshooting HERO

Collecting logs and activating traces

Log files for HERO components are located inside the respective containers.
In case of issues, it is recommended to check the status of the containers. Run the following command:
docker ps -a
If you find containers that are in a non-running or restarting status, download the respective log files.
The list of HERO containers follows:

HERO Container	Description
keycloak	HERO security server. It manages all the user logins and info
keycloak-db	Keycloak container database
hero-elasticsearch	noSQL database for KPIs, prediction data, and runbook information
hero-kibana	Dashboard visual service: displays charts with actual and estimated KPIs
hero-tomcat	Tomcat server that runs HERO war files and exposes the Web UI
hero-prediction	Container managing KPIs prediction, based on a Python process
hero-nginx	HERO reverse-proxy container, managing all the external connections to HERO

To extract all the available log files from a container, run the command:
docker logs [container_name]
For additional details and options, see Docker documentation.

Usually, the log files extracted from each container will be sufficient to troubleshoot the component issues.
For hero-prediction container, for a more detailed problem determination you can run the additional command:
docker exec hero-prediction cat out_WA.txt (you can run this command only if the container is running)

To download the log files from containers, see Docker documentation.

To **activate HERO traces**, run the following steps:

1. From the directory where HERO has been installed, type docker ps
2. Identify the Tomcat container hero-tomcat and type **docker exec -it hero-tomcat bash**
3. Run the following command: **apt-get install vim**
4. Go to /deployments/apache-tomcat/webapps/Dashboard/WEB-INF/classes folder and open **log4j.properties** file
5. If you want to change from DEBUG level to TRACE level, uncomment **log4j.logger.com.hcl=DEBUG** removing the #, and write TRACE instead of DEBUG
6. From the directory where HERO has been installed, type **docker restart hero-tomcat**

Troubleshooting HERO

Symptom
The KPI dashboard is not displayed.
Cause and solution
Verify that the OpenSearch Dashboards service is up and that the browser can reach it.

Symptom
Queue data is not shown.
Cause and solution
Verify that OpenSearch is up and reachable from the workstation.

Symptom
Environment and workstations are not displayed.
Cause and solution
Verify that the HERO server is up and that the connection to derby instance is working fine.

Symptom

The discovery of a server fails, even with correct user and password.

Cause and solution

1. Check that the user can run SSH on the workstation. From HERO server, run the command:
   ```
   ssh <server_to_discover>
   ```
   and verify that it works.
2. Check if the following message is logged in the Tomcat output: *ERROR SSHConnection:469 - com.jcraft.jsch.JSchException: 4: Received message is too long.*
   In this case, the issue is related to the output written by the .bashrc script. Move any script or instruction that writes output to the .bash_profile.
3. If using dockerized version of WA, check that user can create a docker image. If not, add the user to the "docker" group.
4. If you are discovering a Windows machine, check that all the prerequisites are met. Check that Pyhton has the correct version and is 64-bit.

Symptom

A machine cannot be retrieved.

Cause and solution

If a machine cannot be retrieved, run the following actions:

1. Verify that the connection between HERO and the machine is up and no firewall block is active. To verify the connection, use the **ping** and/or **ssh** command from the HERO machine.
2. Verify that the machine to be retrieved has Workload Automation installed and compatible with HERO version.
3. Verify that the machine to be retrieve is up and running.
4. Remove any login welcome message from the **.bashrc** file.

Symptom

HERO is not starting.

Cause and solution

If the HERO server (Dashboard.war) is not starting, check the following conditions:

1. If the following line shows up in the log:

springSecurityFilterChain' threw exception; nested exception is java.lang.NoClassDefFoundError: javax/xml/bind/JAXBException,
it means that you are trying to start HERO server with Oracle Java/OpenJDK version 9 or 10, which is not supported. You must use either OpenJDK 8 or Oracle Java 8.

2. If you installed HERO on a SELinux machine such as RHEL or CentOS, make sure you set the hostname to the Fully Qualified Domain Name of the machine. To identify it, run the command **hostname --fqdn**.

Symptom

OpenSearch 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 by running the command **ulimit -n**
2. Verify that the number of allowed open files for the current user is at least 65536
3. Check the Hard limit for the current user, by running the command **ulimit -n -H**
4. Check the Soft limit for the current user, by running the command **ulimit -n -S**
5. In case the value of Hard or Soft limit is lower than 65536, increase its value, 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:

"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

OpenSearch Dashboards tool is not creating the default index pattern.

Cause and solution

Delete the OpenSearch Dashboards index (by putting the command “DELETE.kibana” inside OpenSearch Dashboards dev tools) and try to create the index pattern again.

Symptom

The discovery process shows the following warning message: The machine has been discovered, with some warnings - No such file: It was not possible to establish an http callback via CURL command. The monitors will not work. Check that is possible to send an http/https request from the machine you are discovering to HERO server.

Cause and solution

Login to the target machine and run the command: curl www.hcl.com.

If you receive the following command output: *curl: (2) Failed initialization,* the problem is related to the configuration of **LD_LIBRARY_PATH** variable.

Correct the **LD_LIBRARY PATH** so that the curl command works correctly. For example, you might need to change the .bashrc and include system libraries path in that library.

Symptom

Monitors do not activate.

Cause and solution

If the monitors do not activate, it means that the monitored server is not connecting back to HERO.

Manually connect to the remote machine being monitored via ssh client and run the monitors manually. Monitors are located under deployPath (usually userHome\<ip/hostname>\HERO\). Check for any error.

Be sure HERO server is reacheable, https port is opened, and curl is working. The https is configurable, so it is the one specified at installation time.

Symptom

While creating OpenSearch Dashboards index pattern to retrieve data from OpenSearch, the following warning message is displayed: Warning - No default index pattern. You must select or create one to continue.

Cause and solution

To solve this problem, run the following CURL commands on the machine that hosts HERO docker installation:

1. docker exec hero-tomcat curl -k -XDELETE "http://elasticsearch:9200/.kibana" -H 'Content-Type: application/json'
2. docker exec hero-tomcat curl -k -XPOST -H "kbn-xsrf: reporting" "http://kibana:5601/api/saved_objects/index-pattern" -H 'Content-Type: application/json' -d' {\"attributes\":{\"title\":\"run*\"}}'
3. docker exec hero-tomcat curl -k -XPOST -H "kbn-xsrf: reporting" "http://kibana:5601/api/kibana/settings/defaultIndex" -H 'Content-Type: application/json' -d' {\"value\":\"run*\"}'

Symptom

Failed to fetch http://archive.ubuntu.com/ubuntu/dists/bionic/InRelease Temporary failure resolving 'archive.ubuntu.com'.

Cause and solution

To solve this problem, run the following steps:

1. edit /etc/default/docker and add the following line:

DOCKER_OPTS="--dns <your_dns_server_1> --dns <your_dns_server_2>"

You can add as many DNS servers as you want to this config.

2. Once saved, restart your Docker service:

sudo service docker restart

Symptom

During HERO online installation, Docker fails to download HERO images from repository.

Cause and solution

If you are using a proxy connection to the internet, the same proxy must be configured in Docker. For details, see: Docker documentation.

Symptom

HERO Web page keeps reloading or blinking.

Cause and solution

Check if keycloak or hero-nginx container log files show "host unreachable" or "unknown host" error messages. The issue might have different causes and solutions:

1. If HERO server is using a proxy connection to the internet, you must configure the same proxy in Docker. For details, see: Docker documentation.
2. Add <hero external port> to your OS firewall. For instance, for RHEL OS run the following commands:
    a. sudo firewall-cmd --zone=public --add-port=<hero external port>/tcp --permanent
    b. sudo firewall-cmd --reload

Symptom

HERO UI login and logout repeatedly

Cause and solution

Ensure that the **secret key** value is the same in the Keycloak UI, in the **ui.properties** file and in the **.tomcat.env** (<HERO_HOME>/CONFIGURATION/HERO) file.

In case of mismatch:

1. Get the **secret key** value from Keycloak UI (Client > nginx > Credentials (tab) > Secret)
2. Update the **ui.properties** file.
3. Remove the configuration volume by running: **docker volume rm <BUILD_DIR>_hero-home**
4. Run **docker-compose up --build -d**

Symptom

Alerts by email are not sent and docker logs the message "Hero-tomcat shows app security exceptions from a gmail account".

Cause and solution

The mail account must be properly configured. See Google account help: Sign in with App Passwords.

Symptom

The **Final job stream** monitor discovers an error even if the Final job stream correctly started at the scheduled time.

Cause and solution

Check if HERO time zone (default is UTC) is different from the WA server time zone.

If it is different, run the following steps:

1. Update HERO time zone by changing the docker-compose.yml file. For details, see step 6 of HERO Installation procedure.
2. From <BUILD_DIR> directory, run the command:
   docker-compose up -d –build

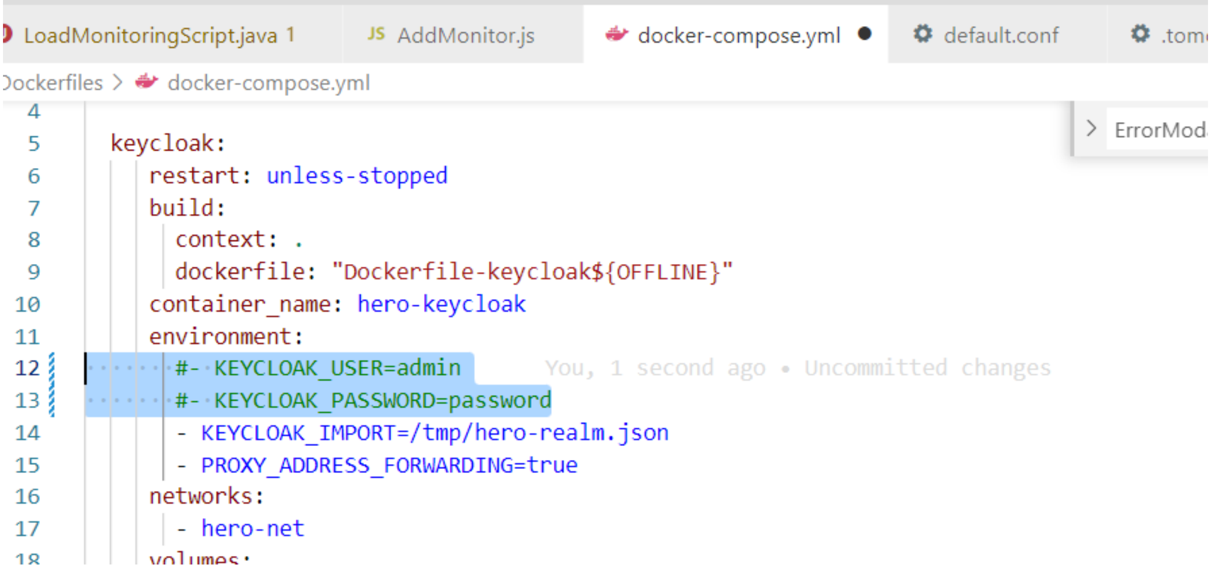
Symptom

After you run the command **docker-compose up**, keycloak container starts to crash, and docker tries to re-launch it in endless loop.

Cause and solution

To solve this issue, run the following steps:

1. Run the command **docker-compose down**.
2. Edit the docker-compose.yml file and comment out the KEYCLOAK_USER and KEYCLOAK_PASSWORD parameters, like in the following example:


LoadMonitoringScript.java 1
AddMonitor.js
docker-compose.yml
default.conf
.tom
ockerfiles > docker-compose.yml
ErrorMod
You, 1 second ago • Uncommitted changes

```
keycloak:
  restart: unless-stopped
  build:
    context: .
    dockerfile: "Dockerfile-keycloak${OFFLINE}"
  container_name: hero-keycloak
  environment:
    #- KEYCLOAK_USER=admin
    #- KEYCLOAK_PASSWORD=password
    - KEYCLOAK_IMPORT=/tmp/hero-realm.json
    - PROXY_ADDRESS_FORWARDING=true
  networks:
    - hero-net
  volumes:
```

3. Run the command **docker-compose up**.

Symptom

When you switch from an environment to another, there is a delay in data refresh on HERO User Interface. .

Cause and solution

Check the SSH connection for all the machines of the environment you want to switch to.