Red Hat Enterprise Linux 9.6

Interacting with the command-line assistant powered by RHEL Lightspeed

Leverage AI-driven expertise of the command-line assistant powered by RHEL Lightspeed to help you configure, manage, and troubleshoot RHEL

Last Updated: 2025-05-30

Red Hat Enterprise Linux 9.6 Interacting with the command-line assistant powered by RHEL Lightspeed

Leverage AI-driven expertise of the command-line assistant powered by RHEL Lightspeed to help you configure, manage, and troubleshoot RHEL

Legal Notice

Abstract

With the command-line assistant powered by RHEL Lightspeed, you can get expert advice and assistance with managing RHEL right from your command line, all by using natural language. The generative AI that powers the assistant incorporates information from the RHEL product documentation and Red Hat Knowledgebase, and can help you to understand, configure, and troubleshoot your RHEL systems

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.
2. Click **Create** in the top navigation bar
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. INTRODUCING RHEL LIGHTSPEED FOR RHEL SYSTEMS

The RHEL Lightspeed intelligent functionalities can help you to manage your system environment in a more accessible way, whether you are less experienced with RHEL or already have experience.

1.1. THE COMMAND-LINE ASSISTANT POWERED BY RHEL LIGHTSPEED

The command-line assistant powered by RHEL Lightspeed is an optional AI tool available within the RHEL command-line interface that includes information from the Red Hat knowledge from Knowledge Centered Service (KCS) articles, RHEL documentation, among other Red Hat resources. You can use the assistant to get help with the following activities, including others:

- Answering RHEL related questions
- Assistance troubleshooting and fixing issues
- Understanding log files
- Asking for recommendations

You can use the command-line assistant powered by RHEL Lightspeed for interactive workflows to solve issues, implement new RHEL features, find information, and more. For example, you can run a command and then use the command-line assistant to help you to understand the output and possible next steps. Or, you can ask a question on SSH, receive suggestions, and ask another question to continue diagnosing the problem.

You can interact with the command-line assistant powered by RHEL Lightspeed by using plain language instead of using complex commands as you might when using a standard command-line interface.

The command-line assistant powered by RHEL Lightspeed does not require direct internet connectivity. This is helpful in cases where, if you do not want to have every RHEL system directly connected to a service over the internet, you can proxy all of the requests from RHEL systems through a single proxy system that is connected to the internet.

RHEL Lightspeed command-line assistant follows the RHEL lifecycle. For information on supported versions and the associated policies, see the Red Hat Enterprise Linux Life Cycle for information on supported versions and the associated policies.

IMPORTANT

The command-line assistant does not have direct access to the information about the system it is running on. For example, the assistant is unable to provide answers on the free memory available on the system that it runs. Instead, the command-line assistant responds with information about a command that you can run to determine how much free memory there is.

RHEL Lightspeed use of generative AI functionalities

RHEL Lightspeed uses the WatsonX AI API LLM (Large Language Model). The model is deployed as a SaaS external infrastructure. The LLM model backing the AI-command-line assistant is hosted on the IBM WatsonX AI platform, and uses the IBM WatsonX LLM IBM® Granite™ model.

1.2. HOW THE COMMAND-LINE ASSISTANT PROCESSES YOUR DATA

While using the command-line assistant interface, you input messages that the command-line assistant transforms and sends to the IBM watsonx™ LLM provider you have configured for your environment. These messages can contain information about aspects of your environment.

Do not enter information into the command-line assistant interface that you do not want to send to the LLM provider.

By using the command-line assistant powered by RHEL Lightspeed, you agree that Red Hat may use all of the messages that you exchange with the LLM provider. The feature is not intended to process personal information, and by using the command-line assistant powered by RHEL Lightspeed you agree to not include any personal information while using the command-line assistant. Support for AI features is provided only for the components that are provided by Red Hat.

CHAPTER 2. INSTALLING THE COMMAND-LINE ASSISTANT POWERED BY RHEL LIGHTSPEED

The command-line assistant powered by RHEL Lightspeed is an optional tool available through the official RHEL 9 repositories.

2.1. INSTALLING THE COMMAND-LINE ASSISTANT POWERED BY RHEL LIGHTSPEED

Before you can start using the command-line assistant powered by RHEL Lightspeed, you must install it through the official RHEL repositories. The command-line assistant is supported in the following architectures:

- AMD and Intel 64-bit (x86_64)
- ARM64 (aarch64)
- IBM Z (s390x)
- IBM POWER systems (ppc64)

To access the command-line assistant powered by RHEL Lightspeed, install it by using RHEL repositories. Do not use **pip install command-line-assistant** because Red Hat does not support this installation option.

Prerequisites

- You have a subscribed RHEL system. For more information, see Getting Started with RHEL System Registration documentation.

Procedure

- On your RHEL system, run the following command:

```
$ sudo dnf install command-line-assistant
```

Verification

- Verify that the installation works by running the command-line assistant. For example:

```
$ c "How to install python?"
```

  The output looks similar to the following example:

```
+*+ Asking RHEL Lightspeed

To install python....
```

- Disable the color output in the command-line assistant. For example:

```
$ c “How to install python?” NO_COLOR=1
```

2.2. PROVISIONING THE COMMAND-LINE ASSISTANT TO RHEL DEPLOYMENTS WITH RED HAT SATELLITE

You can install the command-line assistant powered by RHEL Lightspeed to your host registered to Red Hat Satellite. For that, update the **config --** command-line assistant endpoint, so that the RHEL system registered to Satellite can proxy command-line assistant by using the Satellite Server.

Prerequisites

- Your system is registered with Insights for Red Hat Enterprise Linux.
- You are using the command-line assistant on a host registered to a Satellite 6.17+ server or later versions.
- The Satellite Server must be connected to the internet.
- You have enabled the AppStream repository on the host to be able to install command-line assistant.

Procedure

1. Install command-line assistant powered by RHEL Lightspeed on your registered host.

```
$ sudo dnf install command-line-assistant
```

2. Locate and open the **/etc/xdg/command-line-assistant/config.toml** file.
3. In the **config.toml** file, replace the endpoint configuration option to point to your Satellite or Capsule hostname, for example:

```
#The endpoint points to an API server.
endpoint = "https://satellite.example.server.com/api/lightspeed/v1"
```

4. Save the changes in the **config.toml** file.
5. Restart command-line assistant daemon (**clad**) for the changes to be effective::

```
$ sudo systemctl restart clad
```

CHAPTER 3. USING THE COMMAND-LINE ASSISTANT POWERED BY RHEL LIGHTSPEED ON RHEL SYSTEMS

The command-line assistant powered by RHEL Lightspeed is designed to assist both less experienced and experienced users to interact with RHEL by using the command-line interface. The command-line assistant can help you with tasks such as: answering RHEL related questions, assisting with troubleshooting, assisting with deciphering log entries, and many other tasks.

WARNING

Do not rely on the results from AI tools without human review. Always check the AI and LLM-generated responses for accuracy before using the generated suggestions.

3.1. ASKING QUESTIONS TO THE COMMAND-LINE ASSISTANT

To use the command-line assistant powered by RHEL Lightspeed, use the “c” command followed by your questions in quotation marks. For example:

- Ask a question by using the following syntax: c + “question”. For example:

```
$ c “What is RHEL”
```

WARNING

Do not enter the following types of data when using the command-line assistant, because the assistant is not intended to process data such as:

- Personal information
- Business sensitive information
- Confidential information
- System data information

The following are examples of prompts that you can use to interact with the command-line assistant when using your RHEL systems:

Request information on how to troubleshooting a problem:

```
$ c “how to troubleshoot sshd failing to start”
```

```
$ c “how do I find all the files in the /ect that have been modified in the last hour”
```

3.2. ATTACHING A FILE TO YOUR QUESTIONS TO THE COMMAND-LINE-ASSISTANT

You can attach a file to the command-line assistant powered by RHEL Lightspeed. By doing so, the assistant can provide a tailored response based on that file.

For example, if you want to replicate the volume group, logical volumes, and file systems on another system, you can create a file with the storage information, and run the command-line assistant to get information about the required steps to replicate that specific storage partition in another system. For example:

```
$ c --attachment <storage_info>
```

Optionally, use the short version of the **attachment** command. For example:

```
# $ c -a <storage_info>
```

You can also combine the attachment with a question:

```
# $ c --attachment <storage_info> "how can I replicate the storage configuration in another system*
```

3.3. CHECKING YOUR HISTORY INTERACTION WITH THE COMMAND-LINE-ASSISTANT

Access your conversation history with the command-line assistant powered by RHEL Lightspeed.

- Fetch all user history. For example:

  ```
  $ c history --all
  ```

- Access the first conversation from history. For example:

  ```
  $ c history --first
  ```

- Access the last conversation from history. For example:

  ```
  $ c history --last
  ```

- Filter your history conversation to search for a term to retrieve all questions and answers related to that word. For example:

  ```
  $ c history --filter “podman”
  ```

- Clear all the user history:

  ```
  $ c history --clear
  ```

3.4. REDIRECTING A COMMAND OUTPUT TO THE COMMAND-LINE-ASSISTANT

Use the log file that contains information that you want to understand by redirecting that log file output to command-line assistant powered by RHEL Lightspeed.

```
$ cat <log_file.log> | c
```

If the error or log that you have provided to the command-line assistant does not provide enough information, you can combine the redirect output with a question, for example:

```
$ cat <log_file_error.log> | c "how do I solve this?"
```

You can also redirect an output:

```
$ echo "how do I solve this?" | c -a <log_file_error.log>"
```

3.5. ENABLING THE COMMAND-LINE ASSISTANT TO CAPTURE YOUR TERMINAL ACTIVITY

The command-line assistant powered by RHEL Lightspeed has an optional feature that enables you to reference commands that you previously ran.

WARNING

If you add terminal context in a request and there are no previously captured commands, the command will fail. You can only add context from the terminal while the capture mode is enabled.

1. Enable the terminal capture for your current terminal session. For example:

   ```
   $ c shell --enable-capture
   ```

2. Run at least one command before you reference previous commands.
   After you enable the capture, you can reference the previous output of a command that you ran. For example, to reference the last command, run:

   ```
   $ c -w 1
   ```

   To reference the second to last command, run:

   ```
   $ c -w 2
   ```

3. To stop terminal capture, press the following keys on your keyboard:

   ```
   $ Press Ctrl + D
   ```

3.6. SUBMITTING FEEDBACK ABOUT THE COMMAND-LINE ASSISTANT RESPONSES

You can submit feedback about the responses that you receive when you interact with the command-line assistant powered by RHEL Lightspeed:

```
$ c feedback
```

CHAPTER 4. USING THE COMMAND-LINE ASSISTANT TO DEBUG OR TROUBLESHOOT SYSTEM ISSUES

You can use the command-line assistant powered by RHEL Lightspeed to request information on how to troubleshoot the issues that you face on your system.

The following are examples of questions that you can ask to troubleshoot your system. Ask a question by using the following syntax: c + “question”. For example:

- **$ c “how to troubleshoot network errors”**
- **$ c “I cannot access my server with SSH. Can you give me a list of things to troubleshoot?”**
- **$ c "I am failing to start sssd process"**
- **$ c “I need to boot into a different kernel”**
- **$ c “how to troubleshoot SSHD failing to start”**
- **$ c “how do I find all the files in the /etc that have been modified in the last hour”**
- **$ c "I am failing to start sssd process"**

4.1. USING THE COMMAND-LINE ASSISTANT TO TROUBLESHOOT SSHD SERVICE FAILING TO START

The following example shows how to troubleshoot an SSHD service failing to start by using the following command-line assistant features:

- Optional terminal capture feature in the command-line assistant to reference the output of previous commands when interacting with the command-line assistant.
- Piping data into the command-line assistant.

Prerequisites

- You have enabled the command-line assistant.
- You have root access to your system.

Procedure

1. Check the SSHD status and restart it.

```
$ sudo systemctl status ssh
$ sudo systemctl restart ssh
```

2. Enable the optional command-line assistant terminal capture feature:

```
$ c shell --enable-capture
```

3. Use the **-o 1** option to specify to include the output from the last command that was run.

```
$ c -o 1
```

- If you specify the number 2, that references the output from the second previous 2 commands. This is also true for the additional numbers.

- You can also specify a prompt to run with the command and ask "help me understand the output", and reference the output with the error, so that the command my assistant understands that you are asking for more details on what is the error.
  The command-line assistant takes some time to process the request, and provide several possible solutions. In the example, you can use the suggestion to run the **journtalctl -xeu** command, so that you can use **sshd.service** to check the log files.

4. Run that journal CTL command. Add the tail command to get the last 30 lines, pipe that output into the command-line assistant, and add a query to understand the error.

```
$ journtalctl -xeu sshd.service | tail -n 30 | c "here are the logs, please help me understand this"
```

   The command-line assistant checks the log files and indicates some potential issues. typing error in the **config** file.

5. Ask the command-line assistant to generate a command on how to fix this typing error.

```
$ c "what is the command that I can use to change "Porrt "to "Port" in the /etc/ssh/sshd_config file?"
```

   Use the command suggested by the command-line assistant.

6. Run the command suggested by the command-line assistant. For example:

```
$ sed -i s/Porrt/Port/g /etc/ssh/sshd_config
```

   The output found a permission denied error to edit that file. Rerun the previous command as a sudo user.

```
$ sudo sed -i s/Porrt/Port/g /etc/ssh/sshd_config
```

7. Restart the SSHD service and check the status of the SSHD.

```
$ sudo systemctl restart sshd
$ sudo systemctl status sshd
```

4.2. USING THE COMMAND-LINE ASSISTANT TO TROUBLESHOOT SELINUX ISSUES

The following example provides steps to troubleshoot an SELinux problem with the command-line assistant.

Prerequisites

- You have enabled the command-line assistant.

- You have root access to your system.

Procedure

1. On your terminal, enter the following command to list the **httpd** package version that you have installed in your system:

```
$ sudo rpm -qa httpd
httpd-2.4.62-2.fc40.x86_64
```

2. Show the **httpd.conf** file content.

```
$ sudo cat /etc/httpd/conf/httpd.conf
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see<URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
```

3. Query all **httpdq** packages:

```
$ sudo rpm -qa httpdq
```

4. Identify the ports on which the web server accepts incoming requests:

```
$ cat /etc/httpd/conf/httpd.conf | grep Listen
# Listen: Allows you to bind Apache to specific IP addresses and/or
# Change this to Listen on a specific IP address, but note that if
#Listen 12.34.56.78:80
Listen 80
```

5. Restart the **httpd** service:

```
$ systemctl restart httpd

Job for httpd. Service failed because the control process exited with error code.
See “systemctl status httpd.service" and “journalctl -xeu httpd.service” for details.
```

   a. Run the **journalctl** command for more details on the failed service:

```
$ sudo journalctl -xeu httpd.service
```

6. Use the command-line assistant to troubleshoot the issue and ask why the service is failing:

```
$ sudo c “why did httpd fail to start”
```

   One of the suggestions from the assistant is to query audit logs by using the **ausearch** tool, and use the **AVC** and **USER_AVC** values for the message type parameter. For that, run the following command:

```
$ sudo ausearch -m AVC,USER_AVC -ts recent
```

   a. Ask the command-line assistant about the **selinux httpd** port:

```
$ c “selinux httpd port”
```

The assistant advises to use the **sestatus** command to check the current SELinux status and the content of the httpd services with the following command:

```
$ sudo sestatus

SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

b. View the specific SELinux policy for the httpd services by running the following command:

```
$ sudo cat /usr/share/selinux/targeted/contexts/httpd_var_run_t
No such file or directory
```

c. Ask the command-line assistant about contexts.

```
$ c “i don't have a httpd_var_run_t contexts”
```

The command-line assistant takes some time to process the request, then provides several possible suggestions.

d. The assistant says that you might not have context and need to set it with the following command:

```
$ sudo chcon -R -t httpd_var_run_t
```

e. Ask the CLA about the port:

```
$ c “selinux won't let httpd listen on port 12345”
```

f. Try the following suggestion, run the command:

```
$ sudo semage port -a -t httpd_port_t -p tcp 12345
ValueError: Type httpd_port_t is invalid, must be a port type
```

g. Ask the CLA about the error you see in the output:

```
$ c “how do I fix ValueError: Type httpd_port_t is invalid, must be a port type”
```

7. Run the steps provided by the CLA:

```
$  sudo getenforce
Enforcing
$ setenforce 0
$ sudo systemctl restart httpd
$ sudo systemctl status httpd
```

```
$ sudo ls -Z /usr/sbin/httpd
system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd

$ chcon -t httpd_exec_t /usr/sbin/httpd

$ sudo setenforce 1
```

a. Restart the **httpd** service and check the status of **httpd.service**:

```
$ sudo systemctl restart httpd
Job failed

$ sudo systemctl status httpd.service
Failed to start the Apache Server
```

8. Ask the CLA how to enable **httpd** to listen on **port** 12345:

```
$ c “how do I enable httpd to listen on port 12345 selinux”
```

a. Run the command advised by the CLA:

```
$ sudo setsebool -P httpd_can_network_connect=1
```

9. Restart the **httpd** service and check the status of **httpd.service** again:

```
$ sudo systemctl status htppd
$ sudo systemctl restart httpd
Job failed, see journalctl
```

10. Check the **journalctl** service:

```
$ journtalctl -xeu httpd
Output: An ExecStart= process belonging to unit httpd.service has exited.
```

11. Use the output to ask the CLA to troubleshoot:

```
$ c “An ExecStart= process belonging to unit httpd.service has exited.”
```

a. Run the command that the CLA responds with:

```
$ sudo ausearch -m AVC,USER_AVC -ts recent
Output: “avc:  denied  {name_bind}  for pid=7184 comm=”httpd” src=12345
scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:
unreserved_port_t:s0 tclas=tcp_socket permissive=0"
```

b. Copy the output of the previous command:

```
$ sudo c ““avc: denied {name_bind} for pid=7184 comm=”httpd” src=12345
scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:
unreserved_port_t:s0 tclas=tcp_socket permissive=0”
```

c. Run the following command to resolve the error "SELinux is preventing Apache Server (httpd) from binding to port 12345".

```
$ sudo semage port -a -t http_port_t -p tcp 12345
```

12. Restart the httpd service and check the status of **httpd.service**:

```
$ sudo systemctl restart httpd
No error
$ sudo systemctl status httpd.service
```

The server is configured, up and running, and listening on **port 443**, **port 12345**.

CHAPTER 5. TROUBLESHOOTING THE COMMAND-LINE ASSISTANT POWERED BY RHEL LIGHTSPEED

Diagnose and resolve issues when installing and using the command-line assistant powered by RHEL Lightspeed. Note that this is not an exhaustive list.

Could not find the TLS certificate file, invalid path: /etc/pki/consumer/cert.pem

When you ask a question to the command-line assistant and the assistant fails because the TLS certificate file cannot be found because of an invalid path, it could mean that your system is not registered.

To resolve the error, ensure that you register the system. Follow the steps:

Check if the system is registered:

```
# subscription-manager identity
```

If the system is not registered, use the following command to register the system:

```
# subscription-manager register --username <username> --password <password>
```

Communication error with the server

When you try to run the **c** command on your system, it fails with the following error:

```
Communication error with the server:
HTTPSConnectionPool(host='cert.console.redhat.com', port=443): Max retries exceeded with
url: /api/lightspeed/v1/infer (Caused by ProtocolError('Connection aborted.',
PermissionError(13, 'Permission denied'))). Please try again in a few minutes.
```

To resolve the error, restart the command-line assistant daemon (clad) and rerun the **c** command:

```
$ systemctl restart clad
$ c "<Your_question>"
```

Warning when attaching file

When you try to run the **c** command and try to attach a file, you receive a warning:

"Error: The total size of your question and context (478.46 KB) exceeds the limit of 2.00 KB. Trimming it down to fit in the expected size, you may lose some context."

This happens because the assistant has a 2KB limit for the client.

Clad is not reloading the updated certificate correctly

When you try to run **clad** as a non-root user, **clad** does not reload the updated certificate correctly. The certs under **/etc/pki/consumer** need to be owned by root, because it must access a file that is readable only by root.

To resolve the error, change the **key.pem** permission to root and run the **clad** commands as root.

```
$ sudo ls -l /etc/pki/consumer/
$ sudo chown $(whoami):$(id -gn) /etc/pki/consumer/${KEY_NAME}
```

-

CHAPTER 6. APPENDIX: MODIFYING THE CONFIGURATION OF THE COMMAND-LINE ASSISTANT

The command-line assistant daemon (**clad**) is the core of the command-line assistant powered by RHEL Lightspeed that manages the communication with the RHEL Lightspeed services, such as user history management, among other services. The **clad** is a **dbus** activated daemon. Any interaction with the command-line assistant, for example, by entering a **c** command, activates the daemon.

You can modify the command-line assistant configuration, for example, if you use a proxy, or connect to a different database. Note that these configurations are optional.

6.1. SETTING UP A PROXY CONFIGURATION

If you need a proxy for Internet access, you can set up a proxy configuration by making the following changes in the **config.toml** configuration file.

Prerequisites

- The command-line assistant powered by RHEL Lightspeed is installed.

Procedure

1. Access the proxy configuration by opening the **/etc/xdg/command-line-assistant/config.toml** configuration file.
2. Locate and change the following block in the **config.toml** file :

```
# Backend settings for communicating with the external API.
[backend]
...
# proxies = { http = "http://example-host:8002", https = "https://example-host:8002" }
```

3. Uncomment the **proxies** key and define your **http** or **https** proxy host:

```
[backend]
…
# For a https proxy host
proxies = { https = "https://<your-https-proxy-host:1234>"}
```

4. After making the changes, restart **clad** for the changes to be effective:

```
$ sudo systemctl restart clad
```

NOTE

You can use the **http** value and **https** key control if the http or https traffic from **clad** is routed to the specified proxy. However, the protocol does not influence the proxy type selection, and you can have a configuration that uses http proxy for https traffic. For example:

```
https = "http://<your-https-proxy-host:1234>"
```

Additional resources

- For further details about proxy setup syntax, see urllib.request extensible library for opening URLs documentation

Managing databases with the command-line assistant daemon

To store information and give you access to your history database, by default, command-line assistant daemon (**clad**) uses an unencrypted SQLite database. You can install and connect to a different database backend, such as PostgreSQL or MySQL. Clad does not include these databases by default to avoid bringing unwanted dependencies to your system.

6.2. CHANGING THE DEFAULT DATABASE IN THE CONFIGURATION FILE

With the unencrypted SQLite database, you can store information and have access to your history database from the command-line-assistant.

Prerequisites

- You have installed the command-line assistant.

Procedure

1. Install the database of your choice:
   - To install MySQL, enter:

     ```
     # dnf install python3-PyMySQL
     ```

   - To install PostgreSQL, enter:

     ```
     # dnf install python3-psycopg2
     ```

2. Access your database configuration file at **/etc/xdg/command-line-assistant/config.toml**.
3. Locate and comment out the default configuration. For example:

   ```
   [database]
     # type = "sqlite"
     # connection_string = "/var/lib/command-line-assistant/history.db"
   ```

4. Configure the database of your choice. The following information is also available in **/etc/xdg/command-line-assistant/config.toml**.
   a. Set the database type, where ***<db_type>*** can be **mysql** or **postgresql**.
   b. Set the database details.

   ```
   type = <db_type>
   host = "<hostname_or_ip_address>"
   port = "5432"
   username = "<database_user_name>"
   password = "<password>"
   database = "<database_name>"
   ```

5. After changing the database type, restart the **clad** daemon to apply the changes:

```
$ sudo systemctl restart clad
```

6.3. CONNECTING TO YOUR DATABASE BY USING THE STORED SYSTEMD-CREDS PASSWORDS

You can use the **systemd-creds** tool to securely store your encrypted credentials, and use the secrets stored in **systemd-creds** to connect to the database of your choice: PostgreSQL, SQLite, or MySQL.

Prerequisites

- The command-line assistant.
- Access to the database configuration file.

Procedure

1. Access your database configuration file at **/etc/xdg/command-line-assistant/config.toml**.
2. Remove the **username** and **password** parameters from the **[database]** section, for example:

```
[database]
type = "postgresql"
host = "localhost"
port = "5432"
database = "history"
```

NOTE

If you leave the username and password in the configuration file, these credentials take precedence over the **systemd-creds** tool.

3. Generate encrypted credentials for your username or password. The following example uses **systemd-ask-password** commands. The name must follow the schema of **database-username** and **database-password**, otherwise, **clad** does not load the credentials properly.
   a. To generate an encrypted username, run the following command:

```
$ systemd-ask-password -n | ( echo "[Service]" && systemd-creds encrypt --
name=<database-username> -p - - )
>/etc/systemd/system/clad.service.d/<username>.conf
```

   b. To generate an encrypted password, enter:

```
$ systemd-ask-password -n | ( echo "[Service]" && systemd-creds encrypt --
name=<database-password> -p - - )
>/etc/systemd/system/clad.service.d/<password>.conf
```

4. After updating the database credentials, reload **systemd** and restart the **clad** daemon to apply the changes:

```
$ sudo systemctl restart clad
```

CHAPTER 7. FREQUENTLY ASKED QUESTIONS ABOUT USER DATA SECURITY

What system information can the command-line assistant directly access?

The command-line assistant cannot directly access information from your system. For example, if you ask the command-line assistant a question such as "how much free memory does this system have?", the command-line assistant cannot directly gather this information from the system. Instead, the command-line assistant can help you find a command to display how much free memory the system has. You can provide information to the command-line assistant by using various methods, such as the following:

- Including the information in the prompt or question
- Attaching a file to the command-line assistant with the **--attachment** option
- By using a shell pipeline to pass output from a command to the command-line assistant
- By using the **--with-output** option.

What is the process flow for input and output in the command-line assistant?

1. The command-line assistant receives your question as an input.
    a. Command line-assistant logs and stores complete transcripts of conversations that users have with the command-line assistant. This includes the following information:
        - Queries from the user.
        - The complete message is sent to the configured Large Language Model (LLM) provider, which includes system instructions, referenced documentation, and the user question.
        - The complete response from the LLM provider.
2. The input is processed by the backend.
3. The command-line assistant searches for relevant knowledge related to your query.
4. The command-line assistant takes your query, relevant knowledge, and other instructions and sends for AI inference.
5. You receive an output from the command-line assistant as a response.

The command-line assistant is not intended to process personal information, and you agree to not include any personal information in the input.

Note that the interactions that you have with the command-line assistant are logged locally on your system so that you can have access to your chat history. Those interactions are also logged on the service, and may be used to improve Red Hat's products or services.

Additional resources

- See What are Granite models? article