

Install and configure censhare WP BETA - SysAdmin

Learn how to install and configure censhare WP.

Context

- censhare WP requires Keycloak as authentication method. Existing installations can be incorporated.
- RPM packages are used to install the various parts related to censhare WP.
- **systemd** is used to run services that are related to censhare WP.

Prerequisites

- Knowledge how to administrate Keycloak
- Knowledge how to install and update RPM packages
- Knowledge how to administrate services running with systemd

Components

The following components are required for censhare WP:

- censhare Server + database
- Cloud Gateway
- Keycloak
- Static Resource Server

Google Cloud AI with censhare WP

Optionally, you can use Google Cloud AI and install this component:

- censhare Google Cloud AI service

Installation with RPM

censhare WP uses Webpack technology. censhare WP requires Keycloak as authentication. censhare WP requires additional services so that censhare Server, Keycloak, and censhare WP can communicate with each other.

For all components besides the database, censhare provides RPM packages. You can download them from a central censhare RPM repository.

censhare provides the following RPM packages:

- censhare Server (rpm package: censhare-Server)
- Keycloak (rpm package: keycloak)
- Static Resource Server (rpm package: censhare-static-resource-server)
- Cloud Gateway (rpm package: censhare-cloud-gateway)
- Google Cloud AI analysis microservice (rpm package: censhare-google-ai)

Add repository

You can download the RPM package from the following source: <https://rpm.censhare.com/censhare-release-rpm/stable/censhare/2020/1/>

To retrieve the required RPMs, you need access to this RPM repository. If you do not have log-in credentials (name/password), create a support ticket.

Add the source for the RPM packages to the system:

1. Go to **/etc/yum.repos.d/**
2. If no repo file for censhare exists, create one with suffix **repo**, for example **censhare-services.repo**.
3. Add the following lines:

```
[censhare-release]
name=censhare-services
baseurl = https://USER:PASSWORD@rpm.censhare.com/censhare-release-rpm/stable/censhare/2020/1/
enabled=1
gpgcheck=1
gpgkey = https://repos.censhare.com/6CAE093C.pub
```

Install packages

All RPMs are signed with a GPG key to verify your downloads and the repository.

1. Import the key into your system:

```
rpm --import https://rpm.censhare.com/public/6CAE093C.pub
```

2. Install the basic services:

```
yum install censhare-static-resource-server censhare-cloud-gateway
```

3. If you do not have Keycloak installed:

```
yum install keycloak
```

4. If desired, install the Google AI microservice:

```
yum install censhare-google-ai-microservice
```

5. If you do not have a running server instance, install the censhare Server:

```
yum install censhare-server
```

Install database

For more information on installing the database:

- [Installing Oracle Database Using RPM Packages](#)
- [Installing Postgres](#)
- <https://yum.postgresql.org>

Check installed censhare packages

```
yum list *censhare*
```

Check for updates:

```
yum check-update
```

Update a package:

```
yum update PACKAGE_NAME
```

For example:

```
yum update censhare-static-resource-server
```

Update all:

```
yum update
```

Configuration

After installation, configure the following components:

- Keycloak
- Cloud Gateway
- Static Resource Server

censhare Admin Client

1. In the censhare Admin Client, go to **Configuration > Services > Webserver** and double-click **Configuration**.
2. In the Configuration dialog, enable **Service enabled** and click OK.

3. Update the server configuration.

Keycloak for censure WP

If not already done, first you must [configure Keycloak](#). In particular, configure the censure WP client in Keycloak.

From the censure WP client configuration in the Keycloak administration console, make a note of the following information:

- **Keycloak client name**
If you follow the censure Keycloak documentation, the default name is **censure 5 OpenID client**. It may be different in your case.
- **Keycloak Client ID**
If you follow the censure Keycloak documentation, the default name is **censure5**. It may be different in your case.
- **Keycloak client secret**
The censure WP client **Secret** from your Keycloak server
- **Keycloak server base URL**
For example: <http://keycloak..com>

You need this information in the next step for the configuration of Cloud Gateway.

Cloud Gateway

Before you can configure the Cloud Gateway, you must configure Keycloak. See above.

Default configuration settings for the service are set in the code. So you do not need to set the default values in the **application.yml** configuration file. To show this, these settings are commented out. This prevents that the application.yml file overwrites default settings with older values when the settings have been changed in the code. If you do not use the default settings, uncomment the relevant lines and change the settings.

When installing the RPM package, the **application.yml** file is not overwritten if it has changed. Instead, the installation creates an **application.yml.rpmnew** file. Compare both files to detect new or changed default settings.

Adapt the application.yml configuration file for Cloud Gateway:

Have the Keycloak configuration settings mentioned under [Keycloak](#) at hand.

1. Go to **/opt/censure/cloud-gateway**.
2. Open **application.yml**.
3. Set the server port:

```
server.port: 8082
```

4. Enable Load Balancer headers. These are required to use Load Balancer headers and send the right redirect_uri to the Keycloak server:

```
server.use-forward-headers: true
```

Adapt the data for Keycloak:

- **Client ID**

```
spring.security.oauth2.client.registration.keycloak.client-id: <default-value>
```

If you have selected a different censure WP ID in Keycloak than the default, uncomment the line and replace the value accordingly.

- **Client name**

```
spring.security.oauth2.client.registration.keycloak.client-name: <default value >
```

If you have selected a different censure WP client name in Keycloak than the default, uncomment the line and replace the value.

- **Client secret**

```
spring.security.oauth2.client.registration.keycloak.client-secret:
```

Uncomment the line and replace the secret with the respective censure WP client secret in your Keycloak server.

- **External Keycloak URL**

For the following uris, replace the URLs with your external Keycloak URL. For example, <http://keycloak.com>:

```
spring.security.oauth2.client.provider.keycloak.authorization-uri
spring.security.oauth2.client.provider.keycloak.token-uri
spring.security.oauth2.client.provider.keycloak.user-info-uri
spring.security.oauth2.client.provider.keycloak.jwk-set-uri
cg.keycloakLogoutUrl
```

Adjust the routes to the censhare Server for the following IDs:

- censhare5_rest_endpoint
- censhare5_forward_rest_endpoint
- censhare5_upload_endpoint
- censhare5_websocket_endpoint

1. Uncomment all entries for the affected routes.
2. In the uri of each route, change the <http://censhare-server> URLs to point to your censhare Server :

If Cloud Gateway and censhare Server are running on the same computer, you can use the localhost, for example **`http://localhost:9000`**

If Cloud Gateway and censhare Server are not running in the same subnet, use the server name or full server name. For example, server name , full server name **`<censhare-server>..com`**.

Adjust the route to the Static Resource Server for the following ID: static_resources

1. Uncomment the entry for the affected route.
2. Change the uri to point to your Static Resource Server.
If Cloud Gateway and Static Resource Server are running on the same computer, you can use the local host. For example, **`http://localhost:8081`** .
If Cloud Gateway and Static Resource Server are not running in the same subnet, use the server name or full server name. For example, server name , full server name **`<static-resource-server>..com`** .
3. If you have adapted any ports, you find them here:

```
/opt/censer/static-resource-server/application.yml
```

Variable: **server.port**

Restart the service:

```
systemctl restart censhare-cloud-gateway
```

Static Resource Server

Here you can use localhost as hostname only if Cloud Gateway, Static Resource Server, Keycloak, and censhare Server are running on the same computer.

You can use the server name as hostname instead of the full server name if Cloud Gateway, Static Resource Server, Keycloak, and censhare Server are running in the same subnet of your network.

Default configuration settings for the service are set in the code. As of that, you do not need to set the default values in the application.yml configuration file. To show this, these settings are commented out. This prevents that the **application.yml** file overwrites default settings with older values when the settings have been changed in the code. Only uncomment settings if you want to change them.

The installation of the RPM package does not overwrite the application.yml file if it has changed. Instead, the installation creates an application.yml. rpm new file. Compare both files to detect new or changed default settings.

Adapt the application.yml configuration file for the Static Resource Server:

1. Go to **`/opt/censer/static-resource-server`**.
2. Open **application.yml** in an editor.
3. Uncomment the line with **application.server.rest.url** and replace **localhost** with the server name where censhare Server is running. For example, **`http://censhare.your-company.com:9000/ws/rest/`**
4. Uncomment the line with **spring.security.oauth2.resourceserver.jwt.jwk-set-uri** and replace **localhost** with the server name where Keycloak is running. For example, **`http:// authentication.your-company.com:8080/auth/realms/censhare/protocol/openid-connect/certs`**

Webpack

1. Download the latest webpack related to your current censhare Server version.
2. Connect to the server that is running the Static Resource Server via SSH.
3. Create the **/opt/webpack/** folder.
4. Copy the webpack file to that folder via scp.
5. Unpack the webpack file, for example:

```
tar -xf webpack-2020.1.0.tar.gz
```

6. Restart the service:

```
systemctl restart censhare.static-resource-server.service
```

If you want to install the webpack to a different folder, you must uncomment the line with **webserver.content-dir** in **application.yml** in **/opt/censhare/static-resource-server** and change the path to that folder.

Load Balancer

If you use a load balancer, set the following redirects:

- Path **/auth/** to Keycloak (**http://authentication.your-company.com:8080**)
- Path **/login/** to Cloud Gateway (**http://cloud-gateway.your-company.com:8082**)
- Path **/oauth2/** to Cloud Gateway (**http://cloud-gateway.your-company.com:8082**)
- Path **/censhare5/client/** to Cloud Gateway (**http://cloud-gateway.your-company.com:8082**)
- Path **/ws/** to censhare-Server REST (**http://censhare.your-company.com:9000**)

censhare Google Cloud AI service

Optional service.

User login to the censhare web client

Once you have set up all services, users can log in to the censhare web client. They enter their credentials in the Keycloak login screen. You can brand this screen, if you wish. For more information, see [Configure a custom theme for Keycloak](#). They are then logged into censhare Web.

Work with services

Service names:

- censhare Server
- censhare.cloud-gateway.service
- censhare.static-resource-server.service
- keycloak
- censhare.google-ai.service

Check all running services:

```
systemctl status
```

Check running censhare services (does not include Keycloak):

```
systemctl status censhare.*
```

Check for a certain service:

```
systemctl status SERVICE_NAME
```

Start a service:

```
systemctl start SERVICE_NAME
```

Restart a service:

```
systemctl restart SERVICE_NAME
```

Stop a service:

```
systemctl stop SERVICE_NAME
```

For more information, see [How to administrate censhare related systemd services](#).

Monitoring

You find the log files for the service in the **/var/log/censher** directory:

Log files:

- cloud-gateway.log (Cloud Gateway)
- static-resource-server.log (Static Resource Server)
- google-ai.log (censhare Google Cloud AI service)

Result

You know how to install the services for censhare WP. You know how to configure Cloud Gateway and Static Resource Server.

Next steps

Configure censhare Server and database