

censhare WP - Initial configuration

Initial configuration of censhare WP, including Webpack and the required Keycloak settings.

Components to configure

After installation of the required censhare WP components, configure the following:

- Webserver
- Keycloak
- Cloud Gateway
- Static Resource Server
- Enable Webpack (for censhare Web)
- Redirects for load balancer or reverse proxy

Webserver service

By default, Webserver service is activated.

If not, do the following:

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

Keycloak for censhare WP

If Keycloak is already set up for your organization, note down the required information as stated in [Cloud Gateway](#).

If you just installed Keycloak, follow these steps before you continue to configure Cloud Gateway:

1. [Create a Keycloak system admin and start the Keycloak server](#)
2. [Configure the censhare realm and realm keys](#)
3. [Create internal admin user](#)
4. [Configure the censhare web application client](#)
5. [Optionally, configure censhare desktop client](#)
6. [Configure the hosts.xml](#)
7. [Configure censhare Server to access Keycloak](#)

Cloud Gateway

Before you begin



Configure Keycloak first, to carry on with the Cloud Gateway configuration.

To configure Cloud Gateway, you need specific information from Keycloak that you can obtain from the Keycloak administration console. Important to configure here is the **web client** information taken from Keycloak.

Obtain required Keycloak information

1. Log into Keycloak as administrator.
2. Note down the following information from the censhare web application client in the **Clients** section:
 - **Keycloak client name**
If you follow the censhare Keycloak documentation, the default name is **censhare 5 OpenID client**. It may be different in your case.
 - **Keycloak Client ID**
If you follow the censhare Keycloak documentation, the default name is **censhare5**. It may be different in your case.
 - **Keycloak client secret**
The censhare WP client **Secret** from your Keycloak server
 - **Keycloak server base URL**
For example: **http://keycloak:8080/auth**. Note that **https** requires the use of a Reverse Proxy.

The default configuration for the Cloud Gateway service are set in the code. As of that, you do not need to specify the default values in the **application.yml** file. Therefore, these settings are commented out. This prevents the application.yml file from overwriting the default values with older values if the values have been changed in the code. If you do not want to use the default settings, comment out the appropriate 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 for new or changed default settings.

Adapt the data for Cloud Gateway

Have the Keycloak configuration options at hand.

1. Go to **/opt/censer/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.forward-headers-strategy: native
```

Adapt the data for Keycloak

- **Client ID**

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

- **Client name**

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

- **Client secret**

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

Uncomment the line and replace the secret with the respective censhare 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:8080/auth**:

```
spring.security.oauth2.client.provider.keycloak.authorization-uri: https://censhare.example.com/auth/realms/censhare/protocol/openid-connect/auth
spring.security.oauth2.client.provider.keycloak.token-uri: https://censhare.example.com/auth/realms/censhare/protocol/openid-connect/token
spring.security.oauth2.client.provider.keycloak.user-info-uri: https://censhare.example.com/auth/realms/censhare/protocol/openid-connect/userinfo
spring.security.oauth2.client.provider.keycloak.jwk-set-uri: https://censhare.example.com/auth/realms/censhare/protocol/openid-connect/certs
cg.keycloakLogoutUrl: https://censhare.example.com/auth/realms/censhare/protocol/openid-connect/logout
```

Adapt URL for password change

Users need to manage password changes via the Keycloak Account Management. You have to configure the redirect from the password change dialog of censhare Web to the required Keycloak URL.

To make that URL work, it is necessary to configure a rule in Cloud Gateway, which maps it to the Keycloak Account Management page. Add an entry similar to this:

```
spring.cloud.gateway.routes:
- id: censhare5_change_password_redirect
  uri: http://localhost:8080
  predicates:
  - Path=/censhare5/client/change-password
  filters:
  - SetPath=/auth/realms/censhare/account
```

Please adjust <http://localhost:8080> with the correct host and port of your local Keycloak server.

Adapt 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**.

Adapt 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

```
sudo 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.

Important to configure here is the **Auth URLs** taken from Keycloak.

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** 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.rpmnew file. Compare both files to detect new or changed default settings.


Adapt the application.yml 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

The configuration for the webpack files depends on whether you have customized the censhare Web user interface or added additional user interface languages.

Before you begin

 Ensure that **Node.js 10 or higher** is installed.

censhare Web without customization: get webpack file

If you have not yet done any customization of the web interface, you can directly download the webpack file for your censhare WP version from the repository. The download includes the webpaks for German and English as user interface languages.

1. You find the webpack files for the censhare releases here: <https://rpm.censhare.com/artifactory/webpack-release/>.
2. Change to the directory containing your censhare version, for example <https://rpm.censhare.com/artifactory/webpack-release/2021.2.0/>.
3. Download the webpack file, for example webpack-2021.2.0.tar.gz:

```
curl --user repo-user https://rpm.censhare.com/artifactory/webpack-release/2021.2.0/webpack-2021.2.0.tar.gz -o webpack-2021.2.0.tar.gz
```


censhare Web with customizations: create webpack file

The webpack creation is working on git-based and RPM-based installations.

- If you have modified the web interface, you must create your own webpaks. Check the respective sub-directories of **censhare-Custom** directory of the censhare Server if there are modifications of the web interface.
- If you have not changed the web interface but added additional languages for web interface, you must also create your own webpaks.
- Check the **FRONTEND.md** file for more information on the frontend customization:

```
censhare/censhare-Customer/FRONTEND.md
```

For RPM-based installations

 If you are using censhare WP 2021.2.0 or 2021.2.1 on RPM-based installations, you must create the directory:

```
mkdir censhare/censhare-Customer/custom/
```

Otherwise, the webpack creation fails.

1. Go to the **build** directory:
2. Enter:

```
cd ~/censhare/censhare-Client5/web/
```

3. Build webpack:

```
yarn
yarn cs5 release
```

For each language an own webpack is created.

4. Pack the webpaks into one webpack file:

```
cd dist
tar -czf webpack-VERSION.tar.gz censhare5
```

Replace VERSION with the censhare version that you are using.

Deploy the webpack

1. Connect to the server that is running the Static Resource Server via SSH.
2. Create the **/opt/webpack/** folder.
3. Copy the webpack file to that folder via scp.
4. Unpack the webpack file, for example:

```
tar -xf webpack-2021.2.0.tar.gz -C /opt/webpack/
```

5. Restart the service:

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

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.

Redirects

If you use a load balancer such **HAProxy** or a reverse proxy, such as **NGINX**, set the relevant redirects. For more information, see [Load balancing with HAProxy](#).

censhare Google Cloud AI

Optional service. For more information, see [Configure Google Cloud AI service](#).

censhare Social Media

Optional service. For more information, see [Social media management](#).

User login to censhare Web

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. They are then logged into censhare Web.

Next steps

[Check the censhare WP services and logs](#)