



On Premise Configuration



From:

<https://pandorafms.com/manual/!776/>

Permanent link:

https://pandorafms.com/manual/!776/en/documentation/09_pandora_rc/06_pandora_rc_on_premise

2024/06/10 14:34



On Premise Configuration

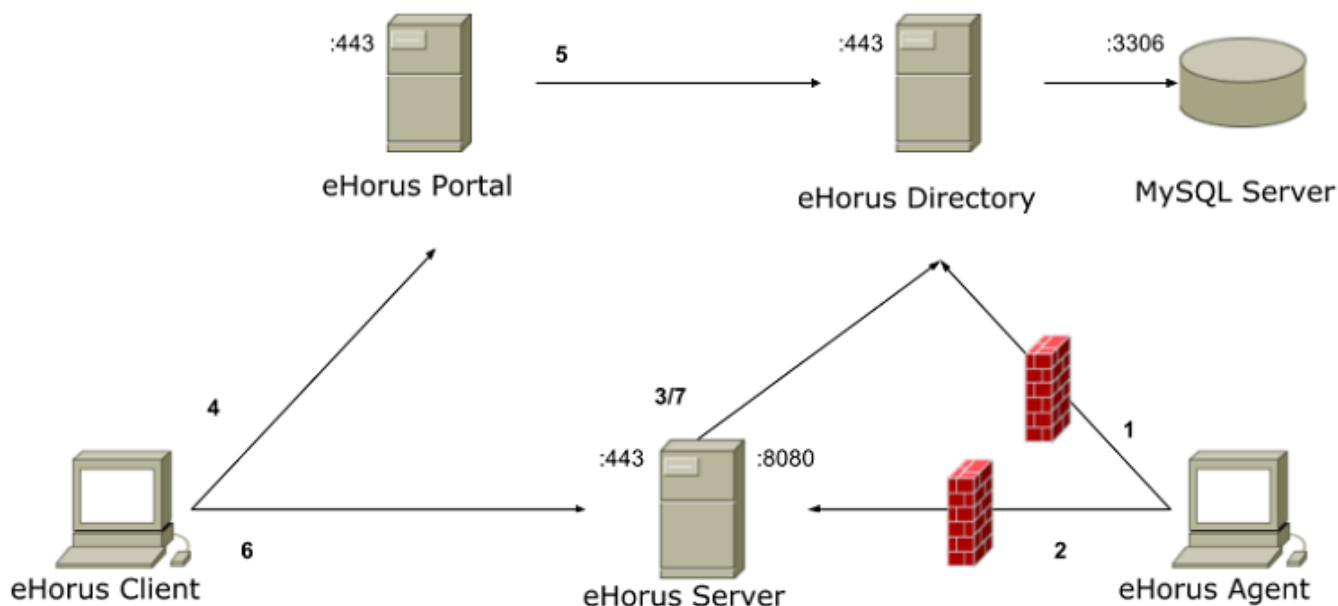
Architecture

This guide will help you to install your own Pandora RC infrastructure. To familiarize yourself with the eHorus application and architecture, see [“Introduction to Pandora RC”](#).

The documentation refers throughout to eHorus, which is the old name of Pandora RC. For compatibility and to minimize errors, paths, file names, configuration files, etc. have not been modified, so the old name will continue to be used at a technical level.

Overview

Below is a general diagram of the architecture and data flow of Pandora RC, as well as the ports on which the different services listen.



- 1 - Provisión y mensajes de actualización de estado.
- 2 - Conexión del agente al servidor de eHorus.
- 3/7 - Autenticación de clientes y agentes a través del API del directorio.
- 4 - Conexión del cliente (navegador) al portal.
- 5 - Llamadas al API del directorio.
- 6 - Conexión del cliente (javascript) al servidor de eHorus.

1. Provisioning and status update messages.
2. Agent connection to Pandora RC server.

3. Client and agent authentication through the directory API (see 7).
 4. Client (web browser) connection to the portal.
 5. Directory API calls.
 6. Client connection (JavaScript) to Pandora RC server.
 7. Client and agent authentication through the directory API (see 3).
- Pandora RC Agent: Software that is installed on the remote computer and allows access to it, available for MS Windows®, GNU/Linux® and Mac OS X®.
 - Pandora RC Server: Manages connections between remote computers (Pandora RC Agent) and users (Pandora RC Client).
 - Pandora RC Client: JavaScript application that connects to the Pandora RC server and allows interaction with the remote computer. It runs through a WEB interface in a browser.



















EHRUS
Terminal
Display
Processes
Services
Files
Share
⏻ ()

Connected to **Travis** (ID: **g98623456**)

Travis

<table style="width: 100%;"> <tr> <td style="width: 15%;">CPU</td> <td>Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 1 Core</td> </tr> <tr> <td>RAM</td> <td>2.99 GB</td> </tr> <tr> <td>HDD</td> <td>73.65 GB</td> </tr> </table>	CPU	Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 1 Core	RAM	2.99 GB	HDD	73.65 GB	<table style="width: 100%;"> <tr> <td style="width: 15%;">OS</td> <td>Windows Windows Server 2012 Standard (64-bit)</td> </tr> <tr> <td>Language</td> <td>en-US</td> </tr> </table>	OS	Windows Windows Server 2012 Standard (64-bit)	Language	en-US
CPU	Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 1 Core										
RAM	2.99 GB										
HDD	73.65 GB										
OS	Windows Windows Server 2012 Standard (64-bit)										
Language	en-US										
<table style="width: 100%;"> <tr> <td style="width: 15%;">Agent version</td> <td>1.0.1</td> </tr> <tr> <td>Uptime</td> <td>2 hours ago</td> </tr> </table>	Agent version	1.0.1	Uptime	2 hours ago	<table style="width: 100%;"> <tr> <td style="width: 15%;">Hostname</td> <td>Travis</td> </tr> <tr> <td>Address</td> <td>192.168.1.100</td> </tr> </table>	Hostname	Travis	Address	192.168.1.100		
Agent version	1.0.1										
Uptime	2 hours ago										
Hostname	Travis										
Address	192.168.1.100										

- Pandora RC Portal: Web application that allows you to manage remote computers, users, groups, and run Pandora RC Client.

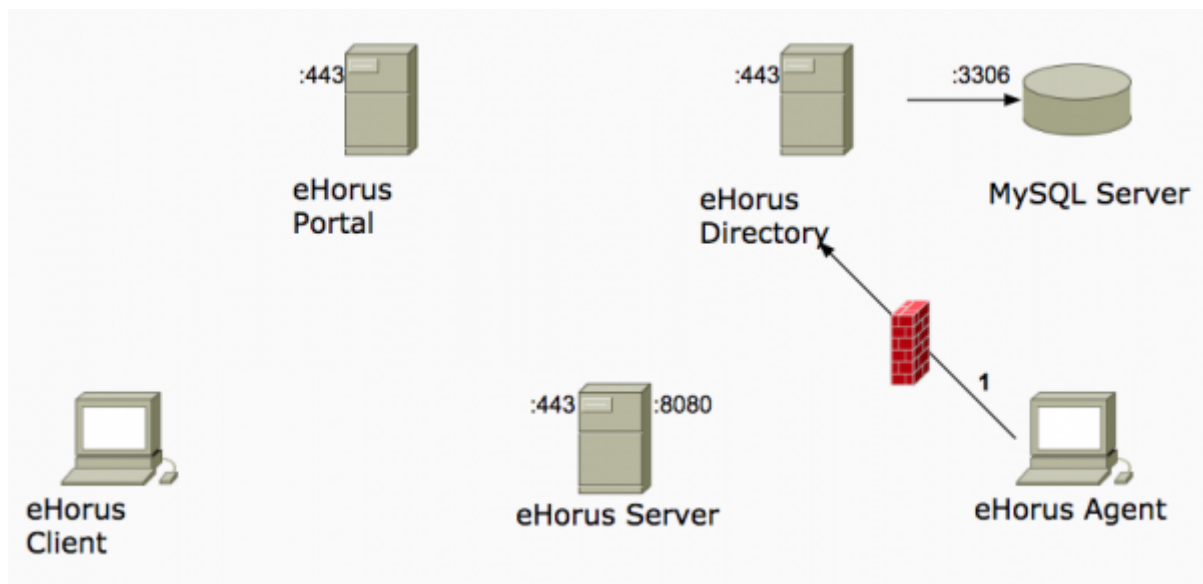
 <p>Last contact: 5 minutes ago</p>  	 <p>Last contact: 3 minutes ago</p>  
 <p>Last contact: 2 minutes ago</p>  	 <p>Last contact: 6 minutes ago</p>  
 <p>Last contact: 4 minutes ago</p>  	 <p>Last contact: 2 minutes ago</p>  

[Refresh](#)
[Check updates](#)

- Pandora RC Directory: Manages application data and access control to remote computers. It communicates with the rest of the elements through a REST API.
- MySQL Server: Stores directory and portal data.

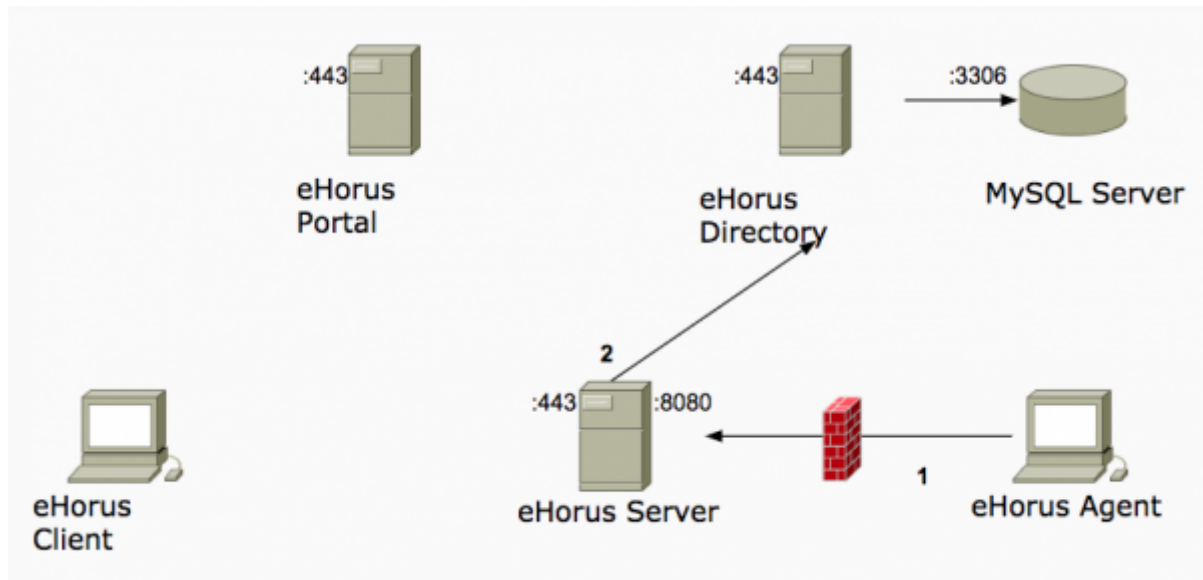
Agent Provision

The agent connects to the directory via HTTPS and receives its eHorus key, an authentication HASH, and a Pandora RC server (see the Server Assignment section) if one has not been specified in the configuration file, that is, if `eh_address` is commented out.



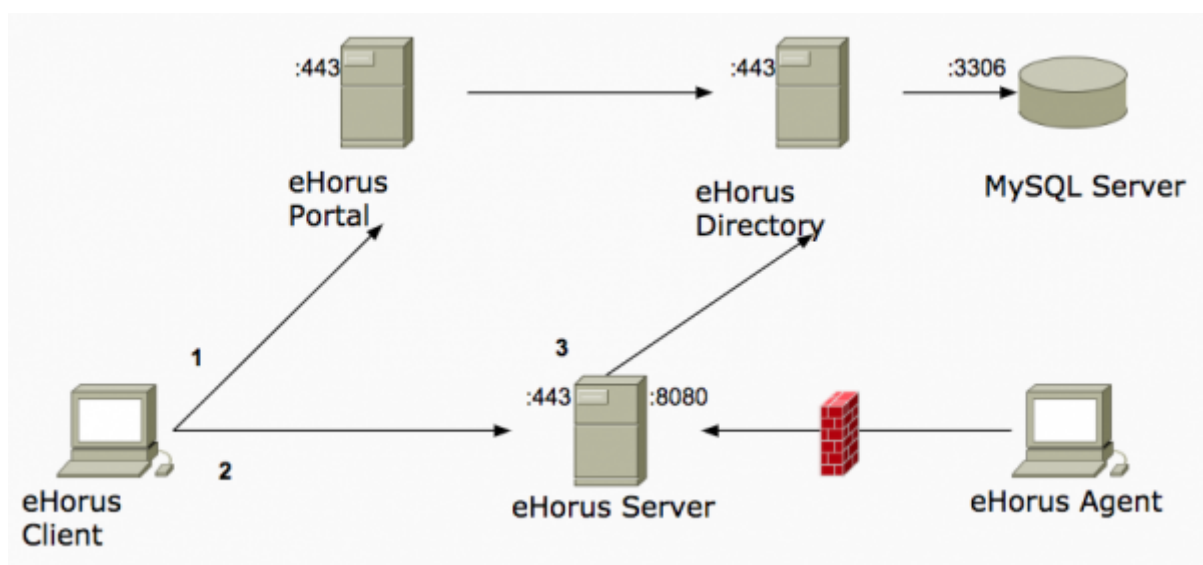
Agent Login

- The agent connects to the Pandora RC server and sends the key and the HASH.
- The server validates the key and HASH through the directory API. If they are not valid, it closes the connection. Otherwise, the agent is associated with the supplied eHorus key.



Client connection

- The user logs into the portal. You select the agent you want to connect to and the browser downloads the client (JavaScript).
- The client connects to the server and sends the agent key and an authentication token provided by the portal.
- The server validates the authentication key and token through the directory API. If they are not valid, it closes the connection. Otherwise, the client connects to the specified Pandora RC agent.

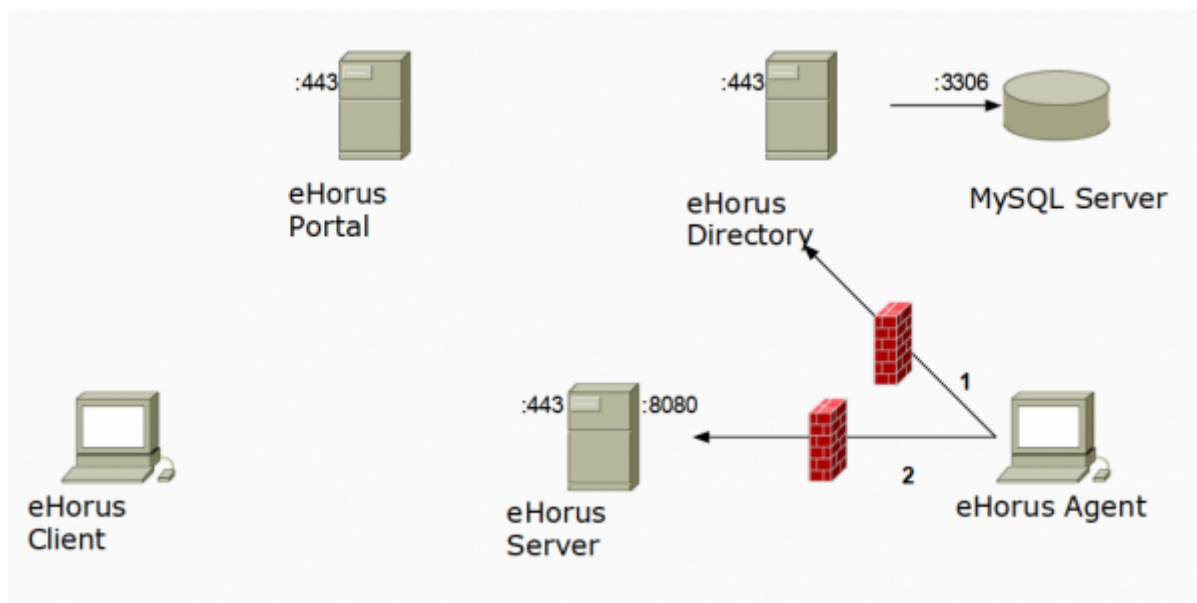


Server Balancing

Server balancing is available from version 1.1.0 of the Pandora RC agent, as long as `eh_balancing` is set to 1 in the configuration file.

- The agent connects to the directory via HTTPS and receives a new server(*) that will replace the current one (see the Server Assignment section).
- The agent connects to its assigned Pandora RC server.

(*) You can receive more than one server to keep a local cache that you will use the next time you need it.



Server Assignment

In both provisioning and balancing, server allocation will be done in one of the following ways:

- By zones: If the user assigned to the agent is assigned a zone, one or several servers will be selected at random that belong to said zone (the zones are labels assigned manually by the administrator, they do not have to correspond to the same zone geographic).
- By geolocation: If the user assigned to the agent has not been assigned a zone, one or several servers will be selected at random that belong to the geographic region closest to the agent (currently, the resolution is at the country level).
- Default server: If the rest of the strategies fail, the server that the administrator has configured by default in the eHorus portal will be assigned.

Prerequisites

Ideally, you will have three hosts to install the Pandora RC directory, portal, and server with at least 4 GB of RAM. If you want you can use an additional host for the directory database.

For environments with less than 100 agents, it is possible to install all the components on a single server as long as you avoid conflicts between the ports of the different components. It is recommended that you have at least 4 GB of RAM.

None of the compPandora RC entities are CPU intensive.

Before starting the installation make sure you have obtained the following files:

```
ehorus-directory-1.0.0.tgz
ehorus-portal-1.0.0.tgz
ehorus-server-1.0.0.tgz
```

Version 1.0.0 will be used as an example, if your version is different adjust the commands shown in this guide as needed. For example:

```
tar zxvf ehorus-directory-1.0.1.tgz
```

instead of:

```
tar zxvf ehorus-directory-1.0.0.tgz.
```

The installation of the different eHorus components will be carried out on a CentOS version 7, Rocky 8 or RHEL 8 system. If you use another operating system, the result may not be as expected.

All commands will be executed as the root user.

You will also need to have received an identification number needed to obtain your Pandora RC license.

In addition, you must have valid X.509 certificates to encrypt communications between the directory, the portal, and the server.

If in doubt, contact your Pandora RC license provider.

Installation of Node.js and NGINX

Pandora RC Directory and Pandora RC Portal need the Node.js environment to run. This guide uses the NGINX web server to access these applications.

On the directory and portal hosts, run the following commands:

```
yum install -y epel-release
yum install -y gcc-c++ make nginx nodejs npm
npm install --global yarn
npm install --global pm2
```


Installing MariaDB

In the directory hosts, run the following commands:

```
yum -y install mariadb-server mariadb
systemctl start mariadb
systemctl enable mariadb
```

Next, we proceed to create the eHorus database:

```
echo "CREATE DATABASE ehorus;" | mysql -u root
```

Then replace the string `STRONG PASSWORD` with a strong password and create the user ehorus:

```
echo "GRANT ALL PRIVILEGES ON *.* TO 'ehorus'@'localhost' \
IDENTIFIED BY 'STRONG PASSWORD' WITH GRANT OPTION;" | mysql -u root
echo "FLUSH PRIVILEGES;" | mysql -u root
```

Finally, set a password for the MariaDB user root (see the MariaDB documentation if in doubt):

```
mysql_secure_installation
```

Installation of the eHorus directory

Run the following commands:

```
mkdir -p /etc/ehorus /var/log/{ehorus-directory,ehorus-directory-clean-db}
```

Unzip the file `ehorus_server_1.0.0.tgz`:

```
tar zxvf ehorus-directory-1.0.0.tgz
```

Install the directory dependencies:

```
yum groupinstall -y 'Development Tools'
yum install -y npm
mv package /opt/ehorus_directory
cd /opt/ehorus_directory
yarn install
```

Create the directory database (you will need the password you used when creating the ehorus user in the database):

```
cd /opt/ehorus_directory
cat db/schema/1-tables.sql | mysql -u ehorus -p ehorus
```

```
cat db/schema/2-rows.sql | mysql -u ehorus -p ehorus
```

Generate two random strings for the directory settings. For example, using the following command:

```
cat /dev/urandom | tr -c -d A-Za-z | fold -w 16 | head -1  
cat /dev/urandom | tr -c -d A-Za-z | fold -w 16 | head -1
```

Create the file `/etc/ehorus/ehorus-directory.pm2.json` with the following content, replacing each instance of the string `'1234567890'` with a different random string generated in the previous step. The value of `JWT_SECRET` will be needed later to install the portal. Replace `USER@DOMAIN` with a valid email address that Pandora FMS will associate with your Pandora RC license:

```
{  
  "apps": [  
    {  
      "name": "ehorus-directory",  
      "script": "server.js",  
      "cwd": "/opt/ehorus_directory",  
      "env": {  
        "NODE_ENV": "development",  
        "PORT": 3000,  
        "JWT_SECRET": "1234567890",  
        "EKID_SECRET": "1234567890",  
        "DB_CONF_PATH": "/etc/ehorus/ehorus-directory.db-config.json"  
      },  
      "env_production": {  
        "NODE_ENV": "production"  
      },  
      "error_file": "/var/log/ehorus-directory/stderr.log",  
      "out_file": "/var/log/ehorus-directory/stdout.log",  
      "merge_logs": true,  
      "min_uptime": "20s",  
      "max_restarts": 20,  
      "max_memory_restart": "200M",  
      "autorestart": true,  
      "restart_delay": 0  
    },  
    {  
      "name": "ehorus-directory-clean-db",  
      "script": "clean-db.js",  
      "cwd": "/opt/ehorus_directory/tools",  
      "env": {  
        "DB_CONF_PATH": "/etc/ehorus/ehorus-directory.db-config.json"  
      },  
      "args": "-i",  
      "error_file": "/var/log/ehorus-directory-clean-db/stderr.log",  
      "out_file": "/var/log/ehorus-directory-clean-db/stdout.log",  
      "merge_logs": true,  
    }  
  ]  
}
```

```
    "min_uptime": "20s",
    "max_restarts": 20,
    "max_memory_restart": "100M",
    "autorestart": true,
    "restart_delay": 1
  }
]
```

Create the file `/etc/ehorus/ehorus-directory.db-config.json` with the following content. Replace the string `STRONG PASSWORD` with the password you used when creating the user `ehorus` in the database:

```
{
  "host": "127.0.0.1",
  "user": "ehorus",
  "password": "STRONG PASSWORD",
  "port": 3306,
  "database": "ehorus",
  "debug": false
}
```

Create the file `/etc/ehorus/ehorus-directory.smtp-config.json` with the following content. Enter the correct parameters for your SMTP server (see the Nodemailer documentation for more information):

```
{
  "debug": false,
  "logger": false,
  "host": "127.0.0.1",
  "port": 465,
  "auth": {
    "user": "USERNAME",
    "pass": "PASSWORD"
  }
}
```

To install the service, run the following command:

```
pm2 start --env production /etc/ehorus/ehorus-directory.pm2.json
pm2 startup
pm2 save
```

Copy the host certificate and public key file to `/etc/ehorus/ehorus_directory.crt` and `/etc/ehorus/ehorus_directory.key` respectively.

To configure NGINX, create the file `/etc/nginx/conf.d/ehorus_directory.conf` with the content shown below. Replace the string `FQDN` with the fully qualified domain name of the host.

Consult the NGINX documentation if you want to customize the configuration:

```
upstream ehorus_directory {
    server 127.0.0.1:3000;
}

server {
    listen 443 ssl;
    server_name FQDN;

    # Add Strict-Transport-Security to prevent man in the middle attacks
    add_header Strict-Transport-Security "max-age=31536000";

    ssl_certificate /etc/ehorus/ehorus_directory.crt;
    ssl_certificate_key /etc/ehorus/ehorus_directory.key;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    ssl_ciphers AES256+EECDH:AES256+EDH:!aNULL;

    location / {
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_pass http://ehorus_directory;
        proxy_redirect off;
    }
}
```

Finally, restart NGINX:

```
service nginx restart
```

Installation of the eHorus portal

Run the following commands:

```
mkdir -p /etc/ehorus /var/log/ehorus-portal
```

Unzip the file ehorus_server_1.0.0.tgz:

```
tar zxvf ehorus-portal-1.0.0.tgz
```

Install the portal dependencies:

```
mv package /opt/ehorus_portal
cd /opt/ehorus_portal
yarn install
yarn run build
```

Generate a random password for the ehorus-api user to create below:

```
cat /dev/urandom | tr -c -d A-Za-z | fold -w 48 | head -1
```

Create the user ehorus-api in the application, which will be used by the portal and the server to make requests to the directory. Replace the string FQDN with the FQDN of the directory and STRONG PASSWORD with the password generated in the previous step:

```
LOGIN_TOKEN=$(curl -s --data "username=admin&password=admin"
'https://FQDN/login' | python -c 'import sys, json; print
json.load(sys.stdin)["token"] ')
curl -s -H "Authorization: JWT $LOGIN_TOKEN" --data "name=ehorus-
api&password=STRONG PASSWORD&fullname=ehorus-api&email=info@pandorafms.com"
'https://FQDN/api/users'
```

On the directory host (this is the only command you will need to run outside of the portal host) run the following command. You will need the password you used when creating the user ehorus in the database:

```
echo 'UPDATE users SET level="admin" WHERE user="ehorus-api"' | mysql -u ehorus
-p ehorus
```

Back at the portal host, generate a JSON Web Token to make requests from the portal to the directory. Replace the string FQDN with the fully qualified domain name of the directory and STRONG PASSWORD with the previously generated password for user ehorus-api:

```
LOGIN_TOKEN=$(curl -s --data "username=ehorus-api&password=STRONG PASSWORD"
'https://FQDN/login' | python -c 'import sys, json; print json.load(sys.stdin)["
token"]')
curl -s -H "Authorization: JWT $LOGIN_TOKEN"
'https://FQDN/api/token?audience=/users&expires=36000d' | python -c 'import sys,
json; print json.load(sys.stdin)["token"]'
```

Create the file /etc/ehorus/ehorus-portal.pm2.json with the following content, replacing __API_SECRET__ with the JSON Web Token generated in the previous step, __JWT_SECRET__ for the value of JWT_SECRET used in the directory installation and __DIRECTORY__ for the full domain of the eHorus directory:

```
{
  "apps": [
    {
      "name": "ehorus-portal",
```

```
"script": "server",
"cwd": "/opt/ehorus_portal",
"send": {
  "NODE_ENV": "development",
  "PORT": 3001,
  "API": "https://__DIRECTORY__",
  "API_SECRET": "__API_SECRET__",
  "JWT_SECRET": "__JWT_SECRET__",
  "MAIL_CONF_PATH": "/etc/ehorus/ehorus-portal.smtp-config.json",
},
"env_production" : {
  "NODE_ENV": "production"
},
"error_file": "/var/log/ehorus-portal/stderr.log",
"out_file": "/var/log/ehorus-portal/stdout.log",
"merge_logs": true,
"min_uptime": "20s",
"max_restarts": 20,
"max_memory_restart": "200M",
"autorestart": true,
"restart_delay": 0
}
]
}
```

Create the file `/etc/ehorus/ehorus-portal.smtp-config.json` with the following content. If you want to receive emails from eHorus, enter the correct parameters in the `smtp` section for your SMTP server (see the [Nodemailer](#) documentation for more information):

```
{
  "from": "PandoraRC <no_reply@localhost.localdomain>",
  "smtp": {
    "debug": false,
    "logger": false,
    "host": "127.0.0.1",
    "port": 465,
    "auth": {
      "user": "USERNAME",
      "pass": "PASSWORD"
    }
  }
}
```

To install the service, run the following command:

```
pm2 start --env production /etc/ehorus/ehorus-portal.pm2.json
pm2 startup
pm2 save
```

Copy the host certificate and public key file to `/etc/ehorus/ehorus_portal.crt` and

/etc/ehorus/ehorus_portal.key respectively.

To configure NGINX, create the file /etc/nginx/conf.d/ehorus_portal.conf with the content shown below. Replace the string FQDN with the fully qualified domain name. Consult the NGINX documentation if you want to customize the configuration:

```
upstream ehorus_portal {
    server 127.0.0.1:3001;
}

server {
    listen 80;
    server_name FQDN;

    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name FQDN;

    # Add Strict-Transport-Security to prevent man in the middle attacks
    add_header Strict-Transport-Security "max-age=31536000";

    ssl_certificate /etc/ehorus/ehorus_portal.crt;
    ssl_certificate_key /etc/ehorus/ehorus_portal.key;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    ssl_ciphers AES256+EECDH:AES256+EDH:!aNULL;

    location / {
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_pass http://ehorus_portal;
        proxy_redirect off;
    }
}
```

Finally, restart NGINX:

```
service nginx restart
```

eHorus server installation

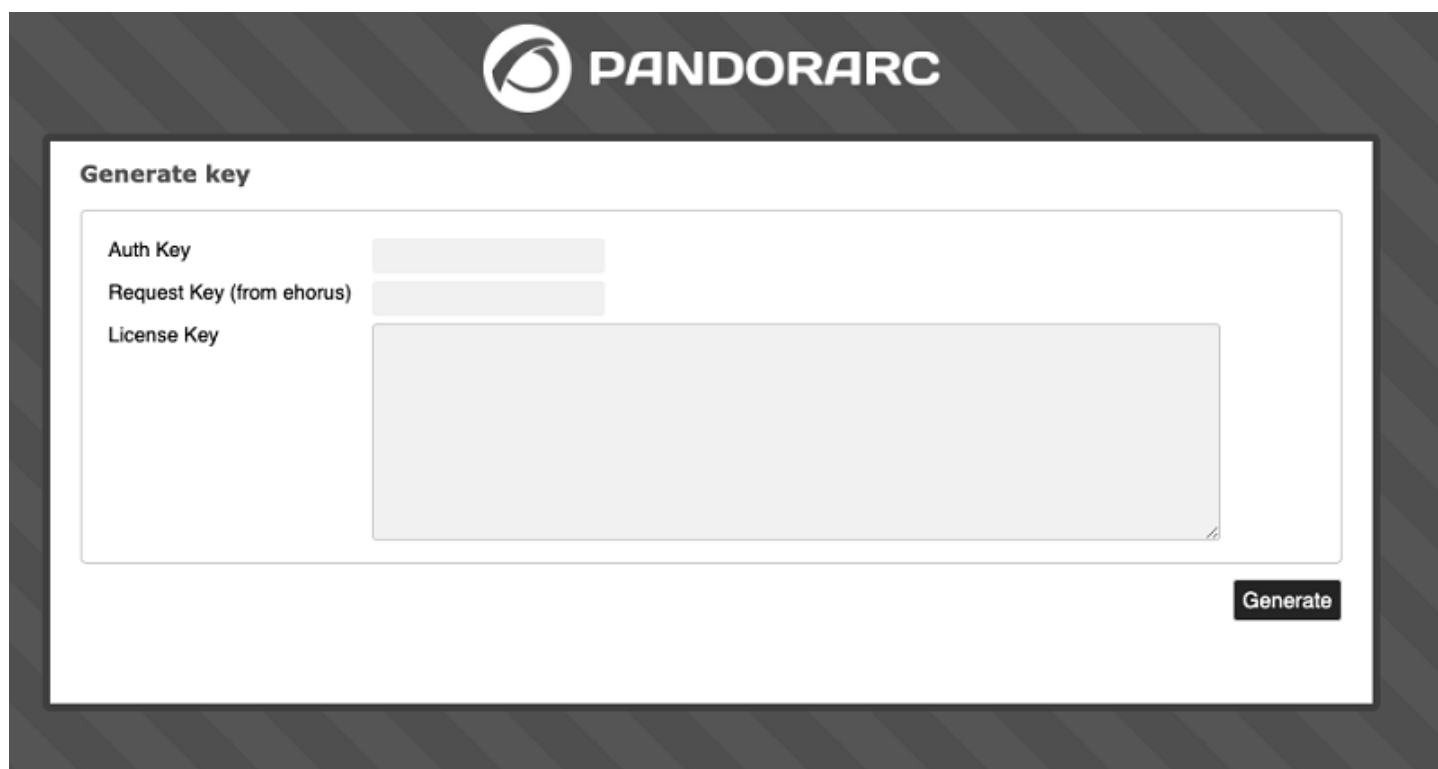
Unzip the file ehorus_server-1.0.0.tgz:

```
tar zxvf ehorus-server-1.0.0.tgz
```

Run the installer:

```
cd ehorus_server-0.1.0  
./ehorus_server_installer --install
```

You will then need to obtain a valid license. To do this, go to [this link](#), enter the identification number you have received from Pandora FMS in the Auth Key field, and in the Request Key (from Pandora RC) field the full domain of the Pandora RC directory. This value must match the domain that appears in the X509 certificate that you install in the directory (for example `directory.pandorafms.com`). Finally, click the Generate button and save the license that will appear in License Key in a safe place.



The screenshot shows a web interface for generating a key. At the top, there is a logo for PANDORARC. Below it, a form titled "Generate key" is displayed. The form contains three input fields: "Auth Key", "Request Key (from ehorus)", and "License Key". The "Auth Key" and "Request Key" fields are small text boxes, while the "License Key" field is a large text area. A "Generate" button is located at the bottom right of the form.

Generate a JSON Web Token to make requests from the server to the directory. Replace the string FQDN with the fully qualified domain name of the directory and STRONG PASSWORD with the password generated during portal installation for user ehorus-api:

```
LOGIN_TOKEN=$(curl -s --data "username=ehorus-api&password=STRONG PASSWORD"  
'https://FQDN/login' | python -c 'import sys, json; print json.load(sys.stdin)["  
token"]')  
curl -s -H "Authorization: JWT $LOGIN_TOKEN"
```



```
'https://FQDN/api/token?audience=/stats&expires=36000d' | python -c 'import sys, json; print json.load(sys.stdin)["token"]'
```

Next, edit the configuration file `/etc/ehorus/ehorus_server.conf` and modify the following parameters:

`license`: Delete the comment character (`#`) and enter the license you obtained in the previous step (do not enclose it in quotes). For example:

```
license 1234567890
```

`ssl_cert`: Absolute path to the server's X.509 certificate. For example:

```
ssl_cert /etc/ehorus/ehorus_cert.pem
```

`ssl_key`: Absolute path to the key file of the X.509 certificate of the servidor. For example:

```
ssl_key /etc/ehorus/ehorus_key.pem
```

`eh_auth_token`: JSON Web Token generated in the previous step. For example:

```
eh_auth_token 1234567890
```

Start the eHorus server manually to verify that the configuration is correct:

```
ehorus_server -f /etc/ehorus/ehorus_server.conf
```

Stop the eHorus server and start it as a service:

```
service ehorus_server start
```

Enter the eHorus portal `https://FQDN` of the portal/ with the user `admin` (password `admin`), and click on Servers:

The screenshot shows the PandoraRC web interface. The top navigation bar includes the PandoraRC logo, a search bar with the text "Enter keywords", and a user profile icon labeled "[admin]". The left sidebar contains a menu with items: Agents, Groups, Users, Updates, Servers (highlighted), Zones, Audit, Setup, and About. The main content area is titled "Servers" and features a search bar. Below the search bar is a table with the following data:

Name	Domain	Address	Actions
moho Default	moho.artica.es	151.80.66.21	

Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom of the main content area, there are pagination controls: "Previous", "1", "Next", and a dropdown menu showing "15". A green "New" button with a plus icon is located at the bottom right of the main content area.

Enter the following information:

- Name: Name of the server. It is merely descriptive.
- Address: Blank. It is not necessary in On Premise installations.
- Domain: Complete domain of the eHorus server.
- Zones: Zone assigned to the eHorus server (optional).

The screenshot shows the PandoraRC web interface with the server configuration form. The top navigation bar and left sidebar are the same as in the previous screenshot. The main content area is titled "Servers" and contains a form with the following fields:

- Name:** Input field containing "moho".
- Address:** Input field containing "151.80.66.21". Below this field is the text "Leave empty if the address cannot be geo located".
- Domain:** Input field containing "moho.artica.es".
- Pandora RC Client port:** Input field containing "443".
- Pandora RC Agent port:** Input field containing "8080".
- Zones:** Dropdown menu with "None" selected.

At the bottom right of the main content area, there are two buttons: "Back" and "Submit".

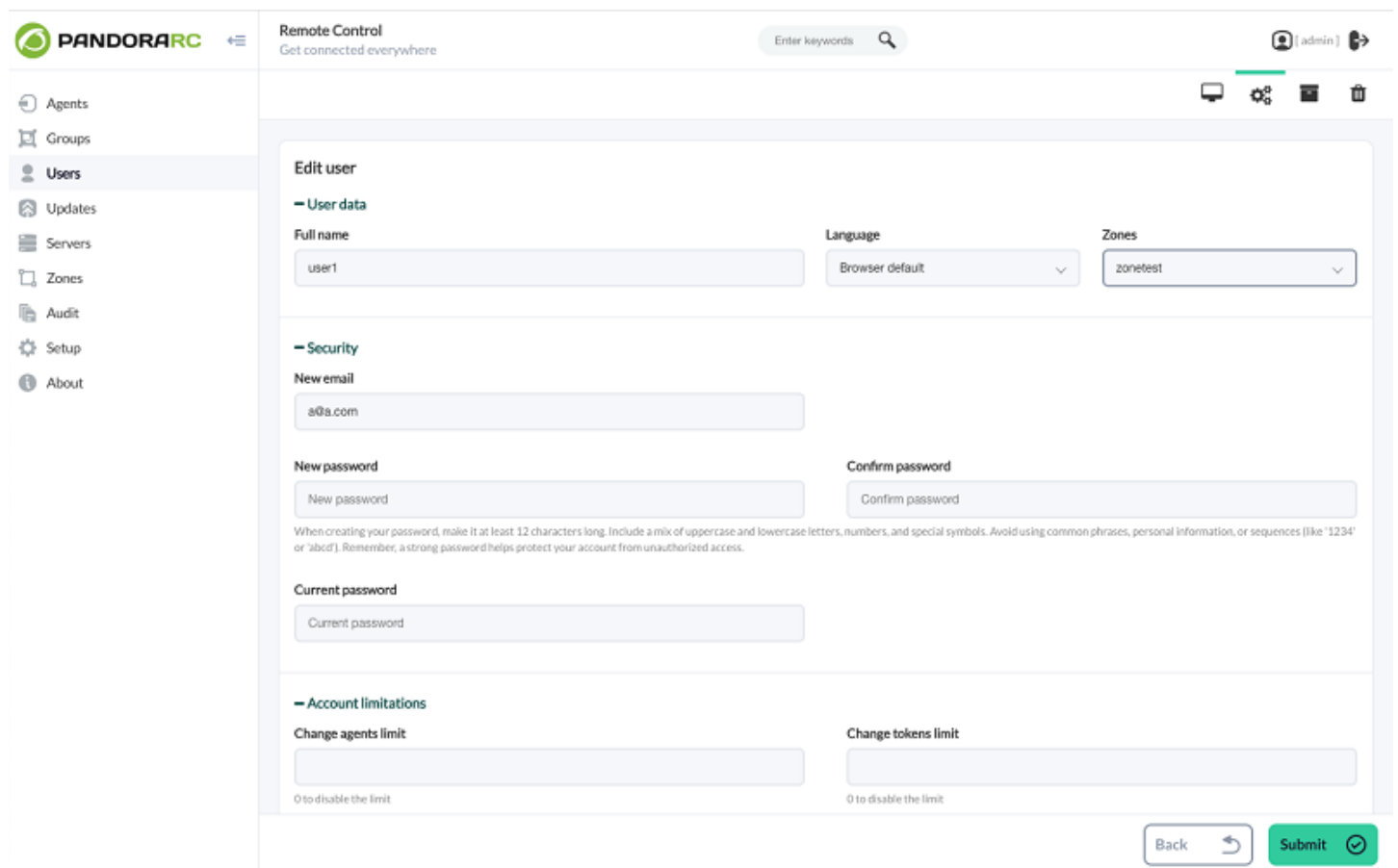
eHorus agents are assigned an eHorus server when they are provisioned (for example, when they

receive a new eHorus directory hash).

As of version 1.1.0, the eHorus agent can request a new server from the eHorus directory if it cannot connect to the server assigned to it (eh_balancing has to be set to 1 in the configuration file). agent configuration).

In both situations, the servers are chosen following the following strategy:

- Balancing based on zones: If the user assigned to the eHorus agent belongs to a zone, a server from said zone is chosen at random.



The screenshot displays the PANDORARC Remote Control interface. The top navigation bar includes the PANDORARC logo, a search bar with the text "Enter keywords", and a user profile icon labeled "admin". A sidebar on the left contains a menu with items: Agents, Groups, Users (highlighted), Updates, Servers, Zones, Audit, Setup, and About. The main content area is titled "Edit user" and is divided into several sections:

- User data**: Contains three input fields: "Full name" (with the value "user1"), "Language" (a dropdown menu set to "Browser default"), and "Zones" (a dropdown menu set to "zonetest").
- Security**: Contains three input fields: "New email" (with the value "a@a.com"), "New password" (with the placeholder "New password"), and "Confirm password" (with the placeholder "Confirm password"). Below these fields is a note: "When creating your password, make it at least 12 characters long. Include a mix of uppercase and lowercase letters, numbers, and special symbols. Avoid using common phrases, personal information, or sequences (like '1234' or 'abcd'). Remember, a strong password helps protect your account from unauthorized access."
- Account limitations**: Contains two input fields: "Change agents limit" and "Change tokens limit", both with the placeholder "0 to disable the limit".

At the bottom right of the form, there are two buttons: "Back" and "Submit".

- Balancing based on geo-location: If the user assigned to the eHorus agent does not belong to any zone, a random server from the closest country is chosen.
- Default server: If all else fails, the server configured as default is returned.

The screenshot shows the PandoraRC web interface. On the left is a navigation menu with items: Agents, Groups, Users, Updates, Servers (highlighted), Zones, Audit, Setup, and About. The main content area is titled 'Remote Control' with the tagline 'Get connected everywhere'. A search bar contains the text 'Enter keywords'. In the top right corner, there is a user profile for 'admin' and several utility icons. The central focus is a card for a server named 'moho' with the following details:

Domain	moho.artica.es - 151.80.66.21
Pandora RC Client port:	443
Pandora RC Agent port:	8080
Country code:	FR

There is a 'Set default' button in the top right corner of the card area.

The zones are managed from the “zones” section of the eHorus portal:



Additional Considerations

- It is recommended to change the password of the admin user in the portal from the Users section.
- The use of firewalls on all hosts is recommended. If you have followed this guide, you will need to access ports 443 of the portal and directory and 443 and 8080 of the eHorus server.

There are two parameters that can be configured in the environment for different purposes:

- Development/production mode: the portal can be configured in two modes: development and production. The first mode is suitable for development and testing tasks, so that the changes made to the portal files will take effect immediately. Under the second mode, changes to portal files will not take effect unless the following commands are executed in order:

```
pm2 delete ehorus-portal
pm2 start --env production /etc/ehorus/ehorus-portal.pm2.json
yarn run build
```

To go to production mode we must execute the following command in the portal:

```
yarn run build
```

- Hash history: this value must be activated for the correct functioning of the environment in case we have the directory and the portal on the same port.

We can activate this feature in the portal configuration file:

```
/etc/ehorus/ehorus-portal.pm2.json
```

To do this we must establish as 0/1 the values corresponding to the configuration token `HASH_HISTORY`, being able to distinguish in each case for development mode and production mode:

```
"env": {
  "NODE_ENV": "development",
  ...
  "HASH_HISTORY": 1
},
"env_production" : {
  "NODE_ENV": "production",
  ...
  "HASH_HISTORY": 1
},
}
```

Once this value has been modified, the command must be executed in the portal:

```
npm run build
```

First steps

Once the infrastructure is installed, you can start installing the eHorus agents on the computers you want to manage remotely. To do so, refer again to [Advanced Settings](#).

Annexes

Portal customization

Keep a backup of your customizations, as they may be lost if you update the software.

Emails

The email templates sent by the eHorus portal are found in the `ehorus_portal/server/mailler/templates` directory. For each email there is a plain text version in the `text` subdirectory, and an HTML version in the `html` subdirectory.

The templates are loaded in memory, so it will be necessary to restart the portal with the command shown below if they are modified:

```
pm2 restart ehorus-portal
```

Templates support macros, enclosed in double braces (for example `{{email}}`), which the eHorus portal will replace with the appropriate value before sending an email.

Welcome screen

- HTML: /opt/ehorus_portal/server/mailer/templates/html/welcome.html
- Plain text: /opt/ehorus_portal/server/mailer/templates/text/welcome.txt

Supported macros:

- {{host}}: Full domain of the eHorus portal.
- {{email}}: Address to which the email is sent.
- {{user}}: Name of the user to whom the email is sent.
- {{name}}: Full name of the user to whom the email is sent.

Password Reset

- HTML: /opt/ehorus_portal/server/mailer/templates/html/password-reset.html
- Plain text: /opt/ehorus_portal/server/mailer/templates/text/password-reset.txt

Supported macros:

- {{host}}: Full domain of the eHorus portal.
- {{email}}: Address to which the email is sent.
- {{user}}: Name of the user to whom the email is sent.
- {{name}}: Full name of the user to whom the email is sent.
- {{token}}: The JSON Web Token used to authorize the password change operation.

Password change

- HTML: /opt/ehorus_portal/server/mailer/templates/html/password-changed.html
- Plain text: /opt/ehorus_portal/server/mailer/templates/text/password-changed.txt

Supported macros:

- {{host}}: The full domain of the eHorus portal.
- {{email}}: Address to which the email is sent.
- {{user}}: Name of the user to whom the email is sent.
- {{name}}: Full name of the user to whom the email is sent.

Email Change

- HTML: /opt/ehorus_portal/server/mailer/templates/html/email-changed.html
- Plain text: /opt/ehorus_portal/server/mailer/templates/text/email-changed.txt

Supported macros:

- {{host}}: The full domain of the eHorus portal.
- {{oldEmail}}: Old email address.
- {{email}}: New email address.
- {{user}}: Name of the user to whom the email is sent.
- {{name}}: Full name of the user to whom the email is sent.

Mail Settings

The configuration for the mailer library should be a JSON file in the form:

```
json
{
  "from": "PandoraRC <no_reply@pandorafms.com>",
  "smtp": {
    SMTP Transport options
  }
}
```

There is a default configuration file in the mailer directory (`ehorus_portal/server/mailer/config.json`). To use another file, the path must be passed using the `MAIL_CONF_PATH` environment variable when starting the `_eHorus Portal_` server.

Assets

If changes are made to the files shown in this section Run the following command to get them copied to the `/opt/ehorus_portal/build` directory:

```
cd /opt/ehorus
npm run build
```

Logo

To change the eHorus portal logo, replace the following files `ehorus_portal/assets/images/logo.png` and `ehorus_portal/assets/images/logo-grey.png`.

CSS

Portal

The eHorus portal CSS files are in the `/opt/ehorus_portal/assets/css` directory.

For the general styles he uses Bulma and for the icons Font Awesome. The fonts are located in the `/opt/ehorus_portal/assets/fonts` directory.

You can also override styles by using a custom CSS file and modifying the `/opt/ehorus_portal/assets/index.html` file to include it.

Customer

The eHorus client CSS files are in the `/opt/ehorus_portal/assets/client/css` directory.

For the general styles it uses Bootstrap and for the icons `xterm.js`. The fonts are located in the `/opt/ehorus_portal/assets/client/fonts` directory.

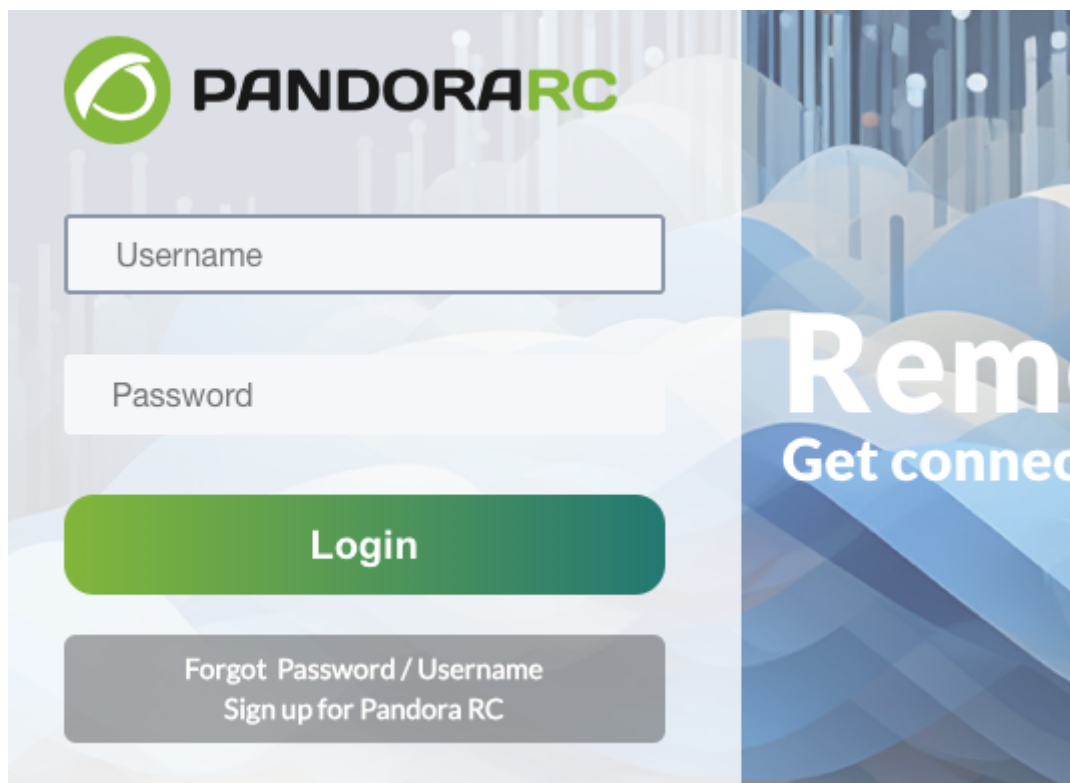
You can also override styles by using a custom CSS file and modifying the `/opt/ehorus_portal/assets/index.html` file to include it.

URL

User registration

The login URL displayed on the login screen can be changed by setting the `URL_CREATE_ACCOUNT` environment variable to the desired URL before executing the `npm run build` command. For example:

```
cd /opt/ehorus_portal  
CREATE_ACCOUNT="https://localhost.localdomain/sign-up" npm run build
```



Certificate generation

If you want to generate your own X.509 certificates, generate the CA certificate first:


```
mkdir/etc/pki/CA/newcerts
mkdir /etc/pki/CA/private
touch /etc/pki/CA/index.txt
echo "01" >> /etc/pki/CA/serial
openssl genrsa -out /etc/pki/CA/private/cakey.pem
openssl req -new -x509 -key /etc/pki/CA/private/cakey.pem -out
/etc/pki/CA/cacert.pem
```

Then install the CA certificate as a trusted root certificate:

```
yum install -y ca-certificates
update-ca-trust force-enable
cp /etc/pki/CA/cacert.pem /etc/pki/ca-trust/source/anchors/
update-ca-trust extract
```

Finally, generate and sign the certificate for eHorus:

```
openssl genrsa -out ehorus.key
openssl req -new -key ehorus.key -out ehorus.req -days 36000
cat ehorus.req ehorus.key > ehorus.pem
openssl ca -out ehorus.crt -in ehorus.pem
```

To prevent the eHorus portal from being shown as an insecure page, consult your browser's documentation to add the certificate of the newly created CA (See <https://wiki.mozilla.org/CA/AddRootToFirefox> and [<https://wiki.mozilla.org/CA/AddRootToFirefox>]).

Other parameters

Agent

proxy_wpad

The eHorus agent allows to automatically discover the proxy settings using the WPAD protocol. To do this, you must specify the URL of the PAC configuration file, or auto to search for said file by DHCP and DNS:

```
proxy_wpad <auto|URL>
```

terminal_export

Semi-comma separated list of environment variable definitions that will be available from the terminal:

```
terminal_export var1="value1";var2="value2";...
```

terminal_path

Semicolon-separated list of directories to be added to the PATH of the terminal:

```
terminal_path dir1;dir2;...
```

Be careful with the `\` character on MS Windows®, as it is used to escape special characters. For example, for `c:\foo\bar` you would write `c: foobar` .

`terminal_script`

Commands that will be sent as is when the terminal is launched. They must form a valid script for the shell in which they are going to be executed (it depends on the OS in which the agent runs):

```
terminal_script <script>
```

`terminal_audit_commands`

If set to `1`, all commands entered by the user will be audited (may include sensitive information such as passwords).

```
terminal_audit_commands 1
```

`access_method_override`

Allows the client to determine the mode of access to the different sections of the agent, regardless of what has been configured with the corresponding `access_*` option (for example, you could specify `always` to prevent the terminal from prompt the user for confirmation if `access_terminal` request has been specified in the agent configuration file). See `extraData.accessMethodOverride` in the client section.

```
access_method_override 1
```

Customer

`extraData.externalID: <ID>`

If this property is defined when instantiating the eHorus client, this identifier will be added to all audit messages generated by the agents to which it connects. It can later be used to filter audit messages.

```
extraData.externalID: 1
```

`extraData.accessMethodOverride`

If this property is defined when instancing the eHorus client and `access_method_override 1` is defined in the agent's configuration file, the client will be able to specify the access mode (always,

request, inform or disable) desired for the various agent services, regardless of what is configured via the `access_*` options.

```
extraData.accessMethodOverride: 'always'
```

API Documentation

All the technical documentation of the API is found in the directory, in the markup file `ehorus_directory/README.md`.