Step-by-step instructions for CentOS 8

Note

These instructions should also work for RHEL 8, AlmaLinux, and Rocky Linux. Some changes may be necessary for RHEL 7 / CentOS 7.

1. Install Node.js

The CARTA controller uses Node.js and supports v12, v14, and v16. Node.js can easily be installed from the CentOS 8 AppStream repository. Here we install v14, as well as the `npm` package manager.

```
# Install Node.js v14:
sudo dnf module enable nodejs:14
sudo dnf install -y nodejs npm

# Check it is installed and working:
node --version
npm --version
```

2. Install MongoDB

The CARTA controller uses MongoDB to store user preferences, etc.. MongoDB is not available through the default CentOS 8 repositories, but we can add a custom repository to install it more easily.

```
# Create a custom MongoDB repo file:
sudo cat <<EOT >> /etc/yum.repos.d/mongodb-org.repo
[mongodb-org-4.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-4.4.asc
EOT

# Install MongoDB:
sudo dnf update
sudo dnf install -y mongodb-org

# Start and enable MongoDB to run on startup:
sudo systemctl start mongod
sudo systemctl enable mongod

# Check that it is working
sudo systemctl status mongod
```

Note

On RHEL7/CentOS7, MongoDB v14 can be installed as follows:

`curl -fsSL https://rpm.nodesource.com/setup_14.x | bash - && yum install -y nodejs`

3. Install the CARTA controller

The easiest way to install the CARTA controller is using `npm`.

```
sudo dnf install -y python3 make gcc-c++
sudo npm install -g --unsafe-perm carta-controller@beta
```

Note

The CARTA controller executable will be installed at `/usr/local/lib/node_modules/carta-controller`. The CARTA frontend will be installed at `/usr/local/lib/node_modules/carta-controller/node_modules/carta-frontend/build`.

Note

Do not pass the `--unsafe-perm` flag to `npm` if using a local install.

Note

On RHEL7/CentOS7 the CARTA controller package can not run with the default gcc version 4.8.5 (there would be an error due to `node-linux-pam`). A work around is to install a newer GCC version from source in order to get a newer `libstdc++.so.6`, then add the location of the newer `libstdc++.so.6` to the LD_LIBRARY_PATH. After that, the CARTA controller can run on RHEL7/CentOS7.

4. Install the CARTA backend

The easiest way may be to install the CARTA backend is from our cartavis RPM repository.

```
# Install the CARTA backend
sudo curl https://packages.cartavis.org/cartavis-el8.repo --output
/etc/yum.repos.d/cartavis.repo
sudo dnf -y install 'dnf-command(config-manager)'
sudo dnf -y install epel-release
sudo dnf -y config-manager --set-enabled powertools
sudo dnf -y install carta-backend-beta

# Check that the backend can run and matches the major version number of the controller
/usr/bin/carta_backend --version
```

5. Install Nginx

The CARTA controller requires a webserver. Here we use NGINX, but Apache should work too.

```
# Install nginx:
sudo dnf install -y nginx
sudo systemctl start nginx
sudo systemctl enable nginx
sudo setsebool -P httpd_can_network_connect 1
sudo firewall-cmd --permanent --zone=public --add-service=http
sudo firewall-cmd --permanent --zone=public --add-service=https
sudo firewall-cmd --reload

# Set up the nginx configuration file using our sample configuration file linked below:
sudo cd /etc/nginx/conf.d/
sudo vi /etc/nginx/conf.d/carta.conf
sudo systemctl restart nginx

# Check it is running:
sudo systemctl status nginx
```

A sample configuration file is provided in the configuration section. This should be adapted to your server configuration.

> **Note**
>
> If there are problems, you can debug with `journactl -xe` and by checking log files in `/var/log/nginx/`.

6. Create the 'carta' user and modify sudoers

For security, we recommend not to run the CARTA controller as the root user. Therefore we create a new user called `carta`.

We will assign the group `carta-users` to every user account and enable them to run `/usr/bin/carta_backend` and the script to close the CARTA backend, `/usr/local/bin/carta-kill-script`, by adding a custom entry to the `sudoers` file.

```
# Create the carta user:
sudo adduser carta
# Check everything is OK
id carta
# It should show 'uid=1000(carta) gid=1000(carta) groups=1000(carta)'

# So that log files can be written:
sudo mkdir -p /var/log/carta
sudo chown -R carta /var/log/carta

# Add the custom sudoers file entry using our sample linked below
sudo visudo -f /etc/sudoers.d/carta_controller
```

An example sudoers configuration is provided in the configuration section.

> **Note**
>
> The only safe way to modify sudoers is using `visudo`. Any syntax errors from directly editing sudoers could make your system unusable.

> **Note**
>
> The `carta` user should not be in the `carta-users` group. `carta-users` should only be assigned to the normal user accounts.

7. Set up the user authentication method

This is the most difficult step and depends on how you authenticate users at your institute. In this step-by-step guide we use PAM local authentication and a local user, `bob`, on the server running the CARTA controller. The user `bob` needs to be part of the `carta-users` group.

With PAM authentication, the `carta` user that runs the CARTA controller requires access to the `/etc/shadow` file in order to authenticate other users. We can enable this by creating a new group called `shadow` and assigning the `/etc/shadow` file to that group.

> **Note**
>
> Only PAM with local authentication requires `/etc/shadow` access. PAM using LDAP, and Google OAuth, do not require `/etc/shadow` access.

```
# Create the test user 'bob':
sudo useradd -G carta-users bob
sudo passwd bob

# A new group called 'shadow' needs to be assinged to the /etc/shadow file and user 'carta':
sudo groupadd shadow
sudo chgrp shadow /etc/shadow
sudo chmod g+r /etc/shadow
sudo usermod -a -G shadow carta
ls -l /etc/shadow
# It should show permissions as ----r-----. 1 root shadow
# It could be helpful to reboot the server at this point
sudo reboot
```

8. Configure the CARTA controller

Create and fill in the `config.json` using our sample configuration file. Also generate private/public keys as they are used by the CARTA controller to sign/verify/refresh access tokens.

```
sudo mkdir /etc/carta
sudo chown -R carta /etc/carta
vi /etc/carta/config.json

# Generate private/public keys:
cd /etc/carta
sudo openssl genrsa -out carta_private.pem 4096
sudo openssl rsa -in carta_private.pem -outform PEM -pubout -out carta_public.pem
```

Please check the CARTA Configuration Schema for all available options.

9. Check everything is working

Here we switch to the `carta` user and test the CARTA controller with our test user `bob`:

```
su - carta
carta-controller -verbose -test bob
```

If the test is successful, the CARTA controller should be ready to deploy.

10. Start the CARTA controller

```
su - carta
carta-controller
```

Now your users should be able to access your server's URL and log into CARTA.

Optional: Set up the CARTA controller to run with pm2

pm2 is a very convenient tool to keep the CARTA controller service running in the background, and even start it up automatically after a reboot.

```
sudo npm install -g pm2
su -carta
pm2 start carta-controller
```

Please refer to the pm2 documentation for detailed instructions.