



WEBADM INSTALLATION GUIDE (STANDALONE AND HIGH AVAILABILITY SETUPS)

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs Security.

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to info@rcdevs.com.

WebADM Installation Guide (Standalone and High Availability setups)

[Standalone](#) [HA](#) [Cluster](#) [Failover](#)

1. Product Documentation

This document is an installation guide for WebADM Server in standalone and high availability mode. WebADM server is the main component to install and deploy OpenOTP in your environment.

WebADM usage manual is not covered by this guide and is documented in the [RCDevs WebADM Administrator Guide](#).

2. Product Overview

WebADM is a powerful Web-based LDAP administration software designed for professionals to manage LDAP Organization resources such as Domain Users and Groups. It is the configuration interface and application server for RCDevs Web Services and Web Apps such as OpenOTP or TiQR Server.

WebADM can be used standalone, as a powerful LDAP management console. It provides a hierarchical view of LDAP Organizations and many features for managing LDAP users and groups resources. It includes delegated administration (administrators can be created at different levels of the tree structure, with different privileges and views), supports multiple LDAP servers (multi-tenants), Domains, allows multiple authentication modes, provides comprehensive SQL and file-based audit trails, etc...

WebADM is compatible with Novell eDirectory, Microsoft Active Directory 2008 and higher, OpenLDAP, FreeIPA, DS389, Oracle/Sun Directory and [RCDevs Directory Server](#). Other directories might work but are not tested or officially supported by RCDevs.

WebADM comes up with the following embedded components:

- › Apache: Http server for WebADM, its Web services and Web applications.
- › Redis Server: Redis is used for session management and replication in High Availability mode.
- › Watchd Server: Watchd component manages monitoring and failover of services configured with WebADM like LDAP, SQL, Redis, SMTP... if any issue make a service unavailable, Watchd order to all WebADM nodes to switch to another servers.
- › Rsign Server: Public Key Infrastructure service. During the setup, you will be prompted to make WebADM a Subordinate CA of an existing Root/Enterprise CA or to make WebADM as Standalone CA. Both setup can be achieved in Standalone or High Availability deployment and must be considered before the WebADM setup. The default setup is to configure WebADM as a standalone CA.

All these services are mandatory and cannot be separated from WebADM component.

3. System Requirements

The current version of WebADM runs on any 64bit operating systems with GLIBC ≥ 2.5 . The installation package contains the required dependencies allowing WebADM to run on any Linux-based system without another requirement. WebADM only needs an LDAP backend (Novell eDirectory, OpenLDAP, RCDevs Directory Server or Microsoft Active Directory) and a SQL database

(MySQL, MSSQL, MariaDB, PostgreSQL, Oracle, SQLite). Other LDAP and SQL backends might work but are not officially supported. SQL service is generally deployed on the WebADM nodes but can also be deported, same for LDAP services and Radius component. Web Services like OpenOTP and Web applications like Selfdesk must be installed on WebADM nodes.

For running WebADM and its services/applications, as well as the Radius Bridge Server and RCDevs Directory Server, your system should have the requirements explained in the following [server sizing documentation](#):

- > A dedicated server computer or Virtual machine with Linux GLIBC \geq 2.5 (RedHat, CentOS, Debian, Ubuntu, SUSE).
- > Network access with DNS and a working NTP integration.
- > A local or remote LDAP directory server (RCDevs Directory Server, OpenLDAP, Novell eDirectory or Microsoft ActiveDirectory \geq 2008).
- > A local or remote SQL database server (Ex. MySQL, PostgreSQL, Oracle, SQLite).
- > Outbound Internet to <https://cloud.rcdevs.com>
- > An LDAP service account with permissions described in that [documentation](#),
- > An administrative user/group that will manage WebADM and its components with permissions described on that [documentation](#).

4. Preliminary Information

WebADM relies on LDAP and most of the rest of this document is related to LDAP configurations. You should also be familiar with LDAP servers or know the basics of LDAP/AD administration in order to set up WebADM correctly.

Unlike other software, there is no “admin account” to be created in WebADM. Instead, you will log in with your LDAP administrator account in the WebADM Administrator interface. The WebADM administrator account (referred to as Super Admin below) is also generally your existing LDAP server’s administrator account. So the only accounts (admin or user) with WebADM are LDAP accounts.

The configurations described below talk about the WebADM LDAP proxy user and WebADM Administrator accounts. When you log in WebADM, you use an LDAP administrator account. The LDAP permissions and views inside WebADM also correspond to the LDAP permissions (ACLs) as configured and enforced by your LDAP server. This is also an LDAP configuration and not a WebADM configuration.

The WebADM proxy user is a special LDAP account which is used by WebADM to connect the LDAP server by himself (out of an admin session). For example, OpenOTP Server needs to search users and read/write user metadata in the LDAP. The proxy user is used by WebADM for such operations and also need sufficient LDAP permissions to handle these tasks.

LDAP Server1 (RCDevs Directory)

WebADM Freeware Edition v2.0.7
 Copyright © 2010-2020 RCDevs Security, All Rights Reserved

API

Home Admin Create Search Import Databases Statistics Applications About Logout

RCDevs Directory (3)

- dc=WebADM
- o=Demos
- o=Root (6)
 - cn=admin
 - cn=admins_group
 - cn=marcus
 - cn=ppolicy
 - cn=user1
 - cn=user2
- Create / Search Details / Check
- Create / Search Details / Check

Hello Admin (*cn=admin,o=root*)
 Connected as **Super Administrator** to 3bcd8b34772

Support Services

License status: **Valid** (Cloud-based)
 Maintenance included: **No**
 Maintenance mode: **Disabled** ([Enable maintenance mode](#))

Application Status

MFA Authentication Server: **Ok** (v1.5.3)
 Shared Session Server: **Ok** (v1.0.11)
 SMS Hub Server: **Ok** (v1.2.0)
 SSH Public Key Server: **Ok** (v2.0.9)
 Administration Help Desk: **Ok** (v1.0.4)
 OpenID & SAML Provider: **Ok** (v1.4.1)
 Secure Password Reset: **Ok** (v1.1.0)
 User Self-Service Desk: **Ok** (v1.2.0)
 User Self-Registration: **Ok** (v1.2.0)

Configurations Objects

User Domains: **6** ([Details](#)) Mount Points: **3** ([Details](#))
 Client Policies: **2** ([Details](#)) Access Devices: **1** ([Details](#))
 Option Sets: **2** ([Details](#)) Admin Roles: **2** ([Details](#))

Context & Permissions

Administration Level: **Expert**
 Login Context: **o=root** ([Details](#))
 Tree Root Context: **Auto**

Created Objects: **All**
 Allowed Configs: **All**
 Allowed Databases: **All**
 Managed Databases: **All**
 Allowed Logfiles: **All**

Applied Option Sets: **o=root** ([Details](#)) ([Edit](#))

Login Context Options

Unicity Context: **o=root**
 WebADM Quotas: **Disabled**

Figure 1. WebADM Home Page (RCDevs Directory Server)

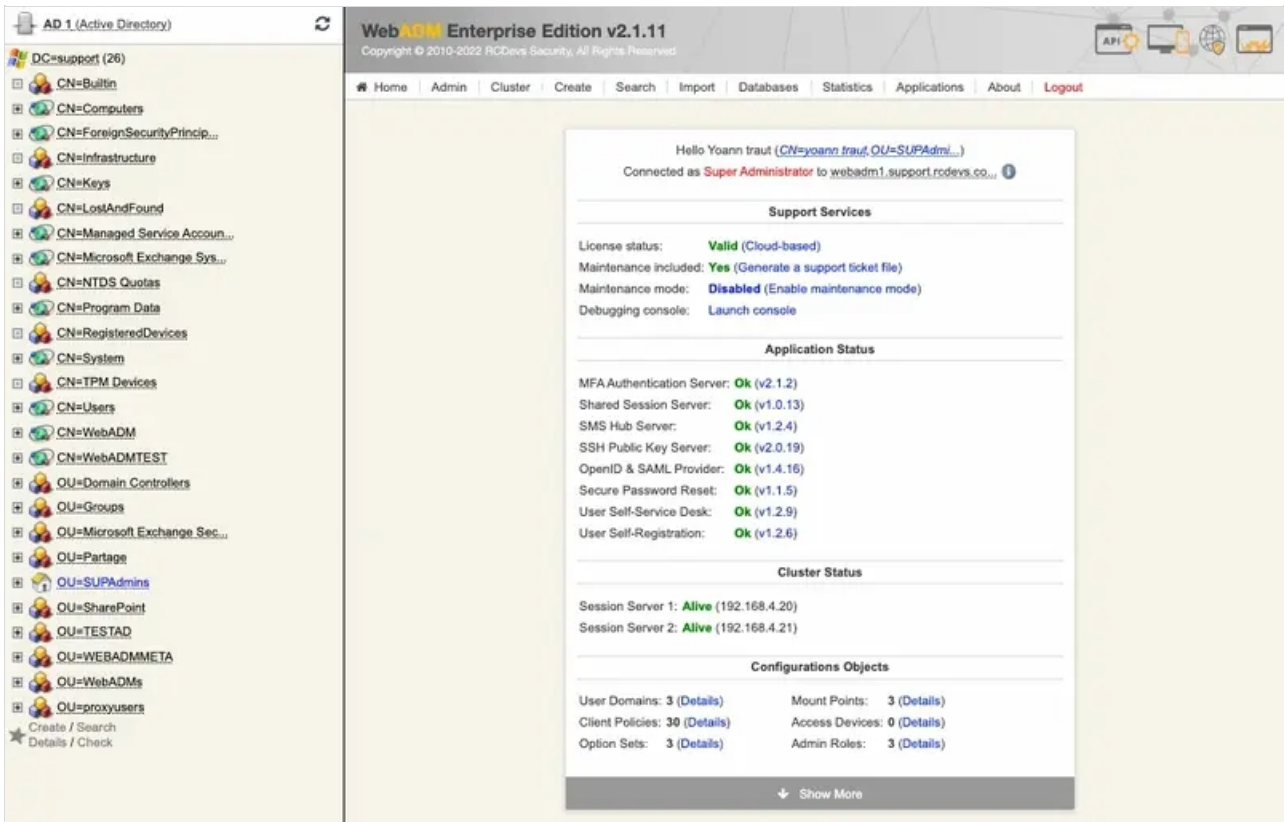
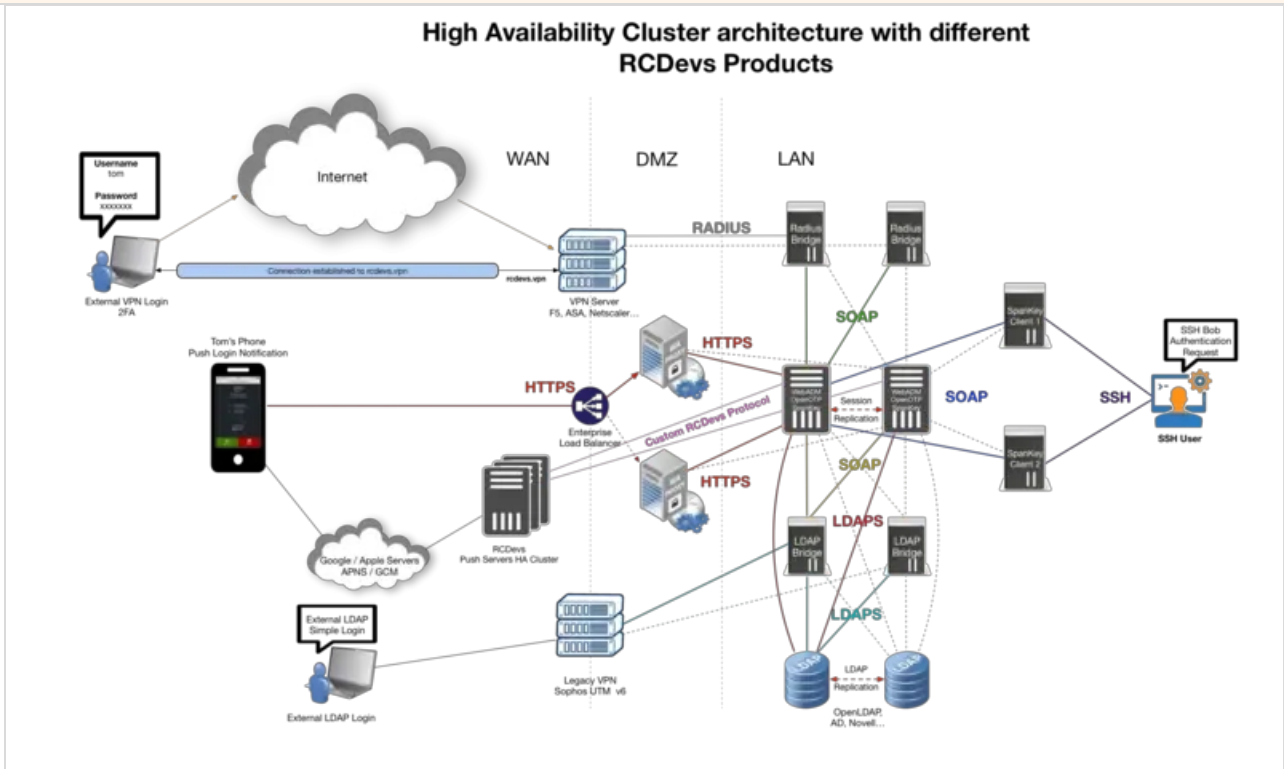


Figure 2. WebADM Home Page (Active Directory)

5. High Availability Mechanisms (enterprise license required)



WebADM supports several high-availability mechanisms for internal and external service failover and for the whole system redundancy. It supports connecting several external data sources such as LDAP directories and SQL databases at the same time and does automatic failover. WebADM connects by default the first declared service (LDAP / SQL / Session Manager / Proxy...) and transparently switches to a secondary service in case of primary service failure.

For systems requiring high-availability and near-zero downtime, WebADM supports cluster setup. In cluster mode, the whole system and services can be deployed on two or more servers for ensuring global redundancy, failover and even load-balancing functionalities. A WebADM cluster is an Active-Active cluster. Any nodes can be contacted at any time.

To enable more than one connection to external services, you just need to configure the external services connections in the `/opt/webadm/conf/servers.xml` configuration file. WebADM will automatically check for service responsiveness in the order the services are specified. It will also connect the first declared service in priority but if this service goes down, it will try to connect the next responsive service. When connected to a non-primary service, WebADM-watchd will re-check if the primary service has recovered every 10 seconds. If at one moment, the service goes up again, WebADM will reconnect its primary service immediately.

The external service switching works for any server connection defined in the `/opt/webadm/conf/servers.xml` file. Failover is done transparently by WebADM and your client systems and end-users won't be affected by the automatic external service switching.

Note

The WebADM session manager and PKI server are specified in the servers.xml file but are local WebADM services (part of the WebADM software). PKI Service (Rsignd) is running on all nodes since WebADM version 2.2.3.

5.1. Connecting Two or more LDAP Servers

In this example, WebADM uses “LDAP Server 1” by default and switches to “LDAP Server 2” in case “LDAP Server 1” goes down.

```
<LdapServer name="LDAP Server 1"  
host="server1"  
port="389"  
encryption="TLS" />  
  
<LdapServer name="LDAP Server 2"  
host="server2"  
port="389"  
encryption="TLS" />
```

It is mandatory that the two LDAP servers use replication. This is automatic with Active Directory when using two domain controllers in the same domain or with Novell eDirectory when LDAP partition replication is set up. RCDevs Directory Server and OpenLDAP require LDAP replication configuration. See below in section 12, the HA configuration of RCDevs Directory.

Remark

Local LDAP connection does not need a security transport layer. Yet, remote LDAP connections should use SSL or TLS if there is a risk of network packet sniffing between the servers.

The LDAP service (Novell eDirectory, OpenLDAP or RCDevs Directory) can be installed on WebADM servers or on other server(s).

5.2. Connecting Two or more SQL Servers

The following example illustrates two redundant SQL servers.

```
<SqlServer name="SQL Server 1"  
type="MySQL"  
host="server1"  
user="webadm"  
password="rwebadm"  
database="webadm" />
```

```
<SqlServer name="SQL Server 2"  
type="MySQL"  
host="server2"  
user="webadm"  
password="rwebadm"  
database="webadm" />
```

SQL databases must be replicated and configured as Master-Master. WebADM must be able to read/write data from every SQL server configured.

5.3. Failover and WebADM Clustering architecture

All the components in WebADM have been designed to support clustering. In this case, the WebADM components (i.e. the WebADM and Radius Bridge software) are deployed on several server computers to provide redundancy, failover or load-balancing. Depending on your cluster usage (failover+load-balancing or failover only), you may configure and use your systems in different manners. The two scenarios explained below are the most common use of WebADM cluster. Yet other configurations are possible, and you may understand in details how WebADM services and connectors work in order to fine-tune your cluster setup.

This is the scenario which corresponds to our previous example. Both WebADM servers, Web services, WebApps can be used at the same time. The remote services (LDAP servers and SQL servers) should be used in the same order by both servers, and they need to be replicated. Unless the LDAP servers use a real-time replication, it is required to use one (and the same) server at a time. Else the user data on the LDAP store could become inconsistent on the different nodes of your cluster during the LDAP replication delay.

The session management services must be used in the same order too. This is required for session sharing and cluster-level operation locking since both WebADM servers are supposed to randomly handle client requests at the same time.

5.4. Architecture for a single/multiple sites

This setup is the recommended one when the whole cluster is running on the same site. If a WebADM cluster is going to be split over multiple sites, the latency between the sites must be good in order to not impact WebADM performances. If you want failover of WebADM service or if you have an important load, then you need to deploy 4 WebADM nodes or more, over the sites.

Architectures are described below.

5.4.1. Single Site (2 nodes cluster)

On Server 1

LDAP Servers: LDAP 1, LDAP 2
SQL Servers: SQL 1, SQL 2
Session Manager: Server 1, Server 2
PKI Server: Server 1, Server 2
STMP Server: Server 1, Server 2
Proxy HTTP: Server 1, Server 2

On Server 2

LDAP Servers: LDAP 1, LDAP 2
SQL Servers: SQL 1, SQL 2
Session Servers: Server 1, Server 2
PKI Server: Server 1, Server 2
STMP Servers: Server 1, Server 2
HTTP Proxy Servers: Server 1, Server 2

5.4.2. Single Site (4 nodes or more cluster)

Configuration designed to support very high load performance.

On Server 1

LDAP Servers: LDAP 1, LDAP 2, LDAP 3, LDAP 4
SQL Servers: SQL 1, SQL 2, SQL 3, SQL 4
Session Manager: Server 1, Server 2, Server 3, Server 4
PKI Server: Server 1, Server 2, Server 3, Server 4
STMP Server: Server 1, Server 2...
Proxy HTTP: Server 1, Server 2...

On Server 2

LDAP Servers: LDAP 1, LDAP 2, LDAP 3, LDAP 4
SQL Servers: SQL 1, SQL 2, SQL 3, SQL 4
Session Manager: Server 1, Server 2, Server 3, Server 4
PKI Server: Server 1, Server 2, Server 3, Server 4
STMP Server: Server 1, Server 2...
Proxy HTTP: Server 1, Server 2...

On Server 3

LDAP Servers: LDAP 1, LDAP 2, LDAP 3, LDAP 4
SQL Servers: SQL 1, SQL 2, SQL 3, SQL 4
Session Manager: Server 1, Server 2, Server 3, Server 4
PKI Server: Server 1, Server 2, Server 3, Server 4
STMP Server: Server 1, Server 2...
Proxy HTTP: Server 1, Server 2...

On Server 4

LDAP Servers: LDAP 1, LDAP 2, LDAP 3, LDAP 4
SQL Servers: SQL 1, SQL 2, SQL 3, SQL 4
Session Manager: Server 1, Server 2, Server 3, Server 4
PKI Server: Server 1, Server 2, Server 3, Server 4
STMP Server: Server 1, Server 2...
Proxy HTTP: Server 1, Server 2...

ldap_routing setting of webadm.conf file can be enabled in order to route the LDAP requests and not overload the primary LDAP server.

Contact RCDevs Service Team for higher design.

5.4.3. Primary and Disaster Recovery sites (x nodes cluster)

As all infrastructures, integrations and needs are different, involve RCDevs Service Team is highly advised to study the topology of your network, your integrations and software architecture/configuration.

5.4.4. Multiple Sites (x nodes cluster and x sites)

Contact RCDevs Service Team for that kind of design. Many things needs to be considered to have a working setup like sites latencies, services replications, components availabilities and so on.

5.5. WebADM Internal Components

A WebADM server includes several internal components. These components are local TCP/IP network services (just like the external services) started by the WebADM startup script and part of the base installation. They must be correctly configured for working in cluster mode.

The HTTP server

The internal Web server provides the SOAP-based web services on port HTTP 8080 and HTTPS 8443. And it provides the Admin Portal and end-user WebApps on HTTPS port 443. SSL server certificates are automatically generated during the initial setup by an internal self-signed certificate authority (CA).

In cluster mode, all the services running over SSL/TLS must have certificates issued by the internal certificate authority. Only one

cluster node will handle the role of the certificate authority. It is a requirement that all the HTTPS services which provide authentication based on client certificates, trust the client certificates issued centralized CA.

The session manager server

This component handles all the user sessions initiated by web services such as OpenOTP and the WebApps. Even if multiple session managers can be specified on each node for failover purposes, in cluster mode, only one session manager should be used for all the cluster nodes at one moment. This is required for the cluster session sharing system to ensure clients requests will be handled correctly whatever node is used and to ensure user data integrity remains consistent. The session manager is used by the cluster nodes to communicate internal information too, such as configuration updates. Web services sessions are also shared for the whole cluster so that internal user working data and user locks remain coherent over your cluster service nodes. The WebADM WebApps use the session manager to handle user login sessions too. This has the big advantage that user browser requests can come randomly to any HTTP service node without impacting the system or the client. This is very handy for working with round-robin load-balancers in front of the service nodes.

The PKI server

All WebADM servers are assigned the certificate authority role in cluster installation. It will run the WebADM Rsignd service which provides certificate signing for the local node and for your other cluster node. The PKI is required during the setup of your cluster nodes for generating SSL server certificates and configuring local CA trusts. It is used by the Admin Portal and the WebApps for issuing and renewing administrator and WebApp user certificates too, for electronic signature, applications installations (e.g: WAPRoxy, Spankey client, Radius Bridge...) During the WebADM setup script, you will be prompted to make WebADM a Subordinate CA of an existing Root/Enterprise CA or to make WebADM as Standalone CA. Both setup can be achieved in Standalone or High Availability deployment and must be considered before the WebADM setup. The default setup is to configure WebADM as a standalone CA.

6. Packages Installations

The WebADM installation consists in installing packages through RCDevs repository or self-installer.

The `/opt/webadm/conf/servers.xml` file contains the LDAP, SQL, Redis, SMTP, PKI server configurations.

The `/opt/webadm/conf/webadm.conf` file contains the main WebADM configurations such as WebADM administrators, service account (proxy_user), LDAP containers used by WebADM to store its LDAP configurations, etc...

6.1. Installation Types

RCDevs provides all its packages as self-installer packages which can be downloaded on [RCDevs Website](#) directly. Packages must be copied on servers, uncompressed and installed.

RCDevs also provides repositories based packages for RHEL/CentOS and Debian/Ubuntu. Please, refer to the following documentation to configure [RCDevs repository](#) on your machines.

6.2. Install WebADM and it's Web-Applications/Web-Services

6.2.1. Install with Yum Repository

On a RedHat, CentOS or Fedora system, you can use our [repository](#), which simplifies updates. Once the repository is installed, you can install packages with the following command:

```
yum install <packages>
```

or

```
dnf install <packages>
```

```
yum install webadm_all_in_one
```

or

```
dnf install webadm_all_in_one
```

```
[root@webadm1 ~]# yum install webadm_all_in_one
Failed to set locale, defaulting to C.UTF-8
Last metadata expiration check: 0:01:03 ago on Wed Jun 15 11:58:25 2022.
Dependencies resolved.
```

```
=====
Package                               Architecture      Version
Repository                             Size
=====
```

Installing:		
webadm_all_in_one	noarch	1.0.1-0
rcdevs-stable	2.0 k	
Installing dependencies:		
openid	noarch	1.4.16-2
rcdevs-stable	10 M	
openotp	noarch	2.1.1-2
rcdevs-stable	14 M	
pwreset	noarch	1.1.4-3
rcdevs-stable	1.3 M	
selfdesk	noarch	1.2.8-3
rcdevs-stable	2.7 M	
selfreg	noarch	1.2.5-2
rcdevs-stable	4.3 M	
smshub	noarch	1.2.4-1
rcdevs-stable	1.5 M	
spankey	noarch	2.0.19-1
rcdevs-stable	13 M	
webadm	x86_64	2.1.9-1
rcdevs-stable	172 M	

Transaction Summary

Install 9 Packages

Total download size: 219 M

Installed size: 374 M

Is this ok [y/N]: y

Downloading Packages:

(1/9): pwreset-1.1.4-3.noarch.rpm

2.3 MB/s | 1.3 MB 00:00

(2/9): selfdesk-1.2.8-3.noarch.rpm

2.8 MB/s | 2.7 MB 00:00

(3/9): openid-1.4.16-2.noarch.rpm

4.8 MB/s | 10 MB 00:02

(4/9): smshub-1.2.4-1.noarch.rpm

4.8 MB/s | 1.5 MB 00:00

(5/9): selfreg-1.2.5-2.noarch.rpm

2.7 MB/s | 4.3 MB 00:01

(6/9): openotp-2.1.1-2.noarch.rpm

2.9 MB/s | 14 MB 00:04

(7/9): spankey-2.0.19-1.noarch.rpm

5.2 MB/s | 13 MB 00:02

(8/9): webadm_all_in_one-1.0.1-0.noarch.rpm

26 kB/s | 2.0 kB 00:00

(9/9): webadm-2.1.9-1.x86_64.rpm

9.7 MB/s | 172 MB 00:17

Total

11 MB/s | 219 MB 00:20

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

Preparing :

1/1

Running scriptlet: webadm-2.1.9-1.x86_64

1/9

Installing : webadm-2.1.9-1.x86_64

1/9

Running scriptlet: webadm-2.1.9-1.x86_64

1/9

Please run /opt/webadm/bin/setup.

Running scriptlet: openid-1.4.16-2.noarch

2/9

Installing : openid-1.4.16-2.noarch

2/9

Running scriptlet: openid-1.4.16-2.noarch
2/9

Running scriptlet: openotp-2.1.1-2.noarch
3/9

Installing : openotp-2.1.1-2.noarch
3/9

Running scriptlet: openotp-2.1.1-2.noarch
3/9

Running scriptlet: pwreset-1.1.4-3.noarch
4/9

Installing : pwreset-1.1.4-3.noarch
4/9

Running scriptlet: pwreset-1.1.4-3.noarch
4/9

Running scriptlet: selfdesk-1.2.8-3.noarch
5/9

Installing : selfdesk-1.2.8-3.noarch
5/9

Running scriptlet: selfdesk-1.2.8-3.noarch
5/9

Running scriptlet: selfreg-1.2.5-2.noarch
6/9

Installing : selfreg-1.2.5-2.noarch
6/9

Running scriptlet: selfreg-1.2.5-2.noarch
6/9

Running scriptlet: smshub-1.2.4-1.noarch
7/9

Installing : smshub-1.2.4-1.noarch
7/9

Running scriptlet: smshub-1.2.4-1.noarch
7/9

Running scriptlet: spankey-2.0.19-1.noarch
8/9

Installing : spankey-2.0.19-1.noarch
8/9

Running scriptlet: spankey-2.0.19-1.noarch
8/9

Installing : webadm_all_in_one-1.0.1-0.noarch
9/9

Verifying : openid-1.4.16-2.noarch
1/9

Verifying : openotp-2.1.1-2.noarch
2/9

Verifying : pwreset-1.1.4-3.noarch
3/9

Verifying : selfdesk-1.2.8-3.noarch
4/9

Verifying : selfreg-1.2.5-2.noarch


```
5/9
  Verifying      : smshub-1.2.4-1.noarch
6/9
  Verifying      : spankey-2.0.19-1.noarch
7/9
  Verifying      : webadm-2.1.9-1.x86_64
8/9
  Verifying      : webadm_all_in_one-1.0.1-0.noarch
9/9

Installed:
  openid-1.4.16-2.noarch      openotp-2.1.1-2.noarch  pwreset-1.1.4-3.noarch  selfdesk-1.2.8-
3.noarch  selfreg-1.2.5-2.noarch  smshub-1.2.4-1.noarch  spankey-2.0.19-1.noarch  webadm-2.1.9-
1.x86_64
  webadm_all_in_one-1.0.1-0.noarch

Complete!
```

All the listed packages as been installed successfully.

6.2.2. Install with Debian Repository

On a Debian system, you can use our [repository](#), which simplifies updates. Once the repository is installed, you can install packages with the following command:

```
apt-get install <packages>
```

or

```
apt install <packages>
```

```
apt-get install webadm_all_in_one
```

or

```
apt install webadm_all_in_one
```

```
root@debian1:/# apt install webadm-all-in-one
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openid openotp pwreset selfdesk selfreg smshub spankey webadm
The following NEW packages will be installed:
  openid openotp pwreset selfdesk selfreg smshub spankey webadm webadm-all-in-one
0 upgraded. 9 newly installed. 0 to remove and 32 not upgraded.
```

Need to get 216 MB of archives.

After this operation, 396 MB of additional disk space will be used.

Do you want to continue? [Y/n]

Get:1 http://repos.rcdevs.com/debian/base ./ openid 1.4.16-2 [10.1 MB]

Get:2 http://repos.rcdevs.com/debian/base ./ openotp 2.1.2-1 [13.7 MB]

Get:3 http://repos.rcdevs.com/debian/base ./ pwreset 1.1.5-1 [1364 kB]

Get:4 http://repos.rcdevs.com/debian/base ./ selfdesk 1.2.9-1 [2785 kB]

Get:5 http://repos.rcdevs.com/debian/base ./ selfreg 1.2.6-1 [4477 kB]

Get:6 http://repos.rcdevs.com/debian/base ./ smshub 1.2.4-1 [1544 kB]

Get:7 http://repos.rcdevs.com/debian/base ./ spankey 2.0.19-1 [13.3 MB]

Get:8 http://repos.rcdevs.com/debian/base ./ webadm 2.1.10-1 [169 MB]

Get:9 http://repos.rcdevs.com/debian/base ./ webadm-all-in-one 1.0.1-0 [1090 B]

Fetches 216 MB in 6s (36.1 MB/s)

debconf: delaying package configuration, since apt-utils is not installed

Selecting previously unselected package openid.

(Reading database ... 4473 files and directories currently installed.)

Preparing to unpack .../0-openid_1.4.16-2_all.deb ...

Unpacking openid (1.4.16-2) ...

Selecting previously unselected package openotp.

Preparing to unpack .../1-openotp_2.1.2-1_all.deb ...

Unpacking openotp (2.1.2-1) ...

Selecting previously unselected package pwreset.

Preparing to unpack .../2-pwreset_1.1.5-1_all.deb ...

Unpacking pwreset (1.1.5-1) ...

Selecting previously unselected package selfdesk.

Preparing to unpack .../3-selfdesk_1.2.9-1_all.deb ...

Unpacking selfdesk (1.2.9-1) ...

Selecting previously unselected package selfreg.

Preparing to unpack .../4-selfreg_1.2.6-1_all.deb ...

Unpacking selfreg (1.2.6-1) ...

Selecting previously unselected package smshub.

Preparing to unpack .../5-smshub_1.2.4-1_all.deb ...

Unpacking smshub (1.2.4-1) ...

Selecting previously unselected package spankey.

Preparing to unpack .../6-spankey_2.0.19-1_all.deb ...

Unpacking spankey (2.0.19-1) ...

Selecting previously unselected package webadm.

Preparing to unpack .../7-webadm_2.1.10-1_amd64.deb ...

Unpacking webadm (2.1.10-1) ...

Selecting previously unselected package webadm-all-in-one.

Preparing to unpack .../8-webadm-all-in-one_1.0.1-0_all.deb ...

Unpacking webadm-all-in-one (1.0.1-0) ...

Setting up selfdesk (1.2.9-1) ...

Setting up webadm (2.1.10-1) ...

WebADM Server needs to be configured.

Please run /opt/webadm/bin/setup.

Setting up openid (1.4.16-2) ...

Setting up selfreg (1.2.6-1) ...

Setting up pwreset (1.1.5-1) ...

```
Setting up smshub (1.2.4-1) ...
Setting up openotp (2.1.2-1) ...
Setting up spankey (2.0.19-1) ...
Setting up webadm-all-in-one (1.0.1-0) ...
```

6.2.3. Install Using the Self-Installer

You first need to download and install the WebADM all-in-one software package. You can download the latest package on the [RCDevs Website](#). Download and copy the WebADM-all-in-one self-installer package to your server. You can copy the package file to the server with WinSCP or SCP. Then connect via SSH to your server, uncompress and run the self-installer package with:

```
[root@webadm1 tmp]# gunzip webadm-all-in-one-2.x.x.sh.gz
[root@webadm1 tmp]# sh webadm_all_in_one-2.1.10-x64.sh
WebADM v2.1.10 (x64 bit) Self Installer
Copyright (c) 2010-2023 RCDevs Security SA All rights reserved.

Install WebADM in '/opt/webadm' (y/n)? y
Extracting files, please wait... Ok
Removing temporary files... Ok
Run WebADM setup script now (y/n)? y
```

Select **y** to start the WebADM setup script.

7. Configuration (WebADM Setup script)

From now, the operating system where you installed WebADM and its packages doesn't matter.

WebADM can be configured as a standalone Certificate Authority or as a Subordinate Certificate Authority.

- › If you don't have any Enterprise Certificate Authority in your infrastructure, then configure WebADM as Standalone CA.
- › If you already have a Root/Enterprise Certificate Authority, then you can generate a subCA certificate/key on your existing CA that will be used by WebADM PKI to generate certificates for various purposes. If you want to configure WebADM as a Subordinate Certificate Authority, copy the certificate and the related key in PEM format in `/opt/webadm/pki/ca/`. If not done before running, the setup script will be stopped.

In that setup script you will be prompted to provide the below information:

- › Configure WebADM as Standalone CA or Subordinate CA;
- › LDAP directory IP/DNS name you want to use with WebADM, port and encryption;
- › Tree-base of your LDAP directory (Active Directory setup);
- › Service account distinguished name (proxy_user in RCDevs jargon);
- › Service account password;
- › User or Group allowed to log in to WebADM Admin Portal.

Take care to have all the required information before starting it.

7.1. Configure the Master node / Standalone server

Run the WebADM setup script with the following command:

```
[root@webadm1 tmp]# /opt/webadm/bin/setup
```

You are then prompted for the WebADM license agreement that you have to accept by entering **Yes**:

RCDEVS WEBADM LICENSE AGREEMENT

RCDevs WebADM Server ("WebADM")

Copyright (c) 2010-2023 RCDevs Security SA, All rights reserved.

IMPORTANT: READ CAREFULLY: By using, copying or distributing the Software Product you accept all the following terms and conditions of the present WebADM License Agreement ("Agreement").

If you do not agree, do not install and use the Software Product.

WebADM includes additional software products provided by RCDevs SA under freeware and commercial licenses. These additional software are installed under the "/opt/webadm/webapps" and "/opt/webadm/websrvs" directories. This Agreement is subject to all the terms and conditions of any such additional software license.

1. **DEFINITIONS.** "Software Product" means RCDevs Server with which the Agreement is provided which may include third party computer information or software, including apache2, php, libmcrypt, libcurl, libgmp, redis, libxml2, libpng, libqrencode, openldap, openssl, apcu, unixodbc, geoip, expat, hiredis, nghttp2, hiredis, libmaxmind, openscn libcouchbase unmodified software and libraries and related explanatory written materials ("Documentation"). "You" means you or any recipient that obtained a copy of the Software Product pursuant to the terms and conditions of the Agreement.

2. **LICENSE.** Subject to your compliance with the terms and conditions of the Agreement, including, in particular, the provisions in Sections 3, 5 and 6 below, RCDevs hereby grants You a non-exclusive and royalty-free license to use and distribute the Software Product solely for non-commercial purposes in worldwide. You may:

- a. download and install the Software Product on any computer in your possession;
- b. use the Software Product and any copy solely for a non-commercial purposes;

c. make any original copies of the Software Product; and

d. distribute any copy of the Software Product only in the form originally furnished by RCDevs with no modifications or additions whatsoever. If You have the slightest doubt that your copy of the Software Product is not original, You must contact RCDevs for an original copy.

3. OBLIGATIONS AND RESTRICTIONS ON LICENSE. The license granted in Section 2 is subject to the following obligations and restrictions:

a. The Software Product and copies are to be used only for non-commercial purposes. Prohibited commercial purposes include, but are not limited to:

(i) Selling, licensing or renting the Software Product to third parties for a fee (by payment of money or otherwise, whether direct or indirect);

(ii) Using the Software Product to provide services or products to others for which you are compensated in any manner (by payment of money or otherwise, whether direct or indirect), including, without limitation, providing support or maintenance for the Software Product;

(iii) Using the Software Product to develop a similar application on any platform for commercial distribution.

You shall use your best efforts to promptly notify RCDevs upon learning of any violation of the above commercial restrictions.

b. RCDevs, in its sole and absolute discretion, may have included a portion of the source code or online documentation of the Software. Except for any such portions, YOU SHALL NOT MODIFY, REVERSE ENGINEER, DECOMPILE, DISASSEMBLE, OR OTHERWISE ATTEMPT TO DISCOVER THE SOURCE CODE OF THE SOFTWARE PRODUCT, except to the extent this restriction is prohibited by applicable law. Further, You may not create derivative works of or based on the Software Product.

c. Any copy of the Software Product that you make must conspicuously and appropriately reproduce and contain RCDevs's copyright and other proprietary notices that appear on or in the Software Product (see Software Product for examples of such notices) and disclaimer of warranty; keep intact the Agreement and all notices that refer to the Agreement and any absence of warranty; and give any other recipients of the Software Product a copy of the Agreement.

d. As used in this Agreement, the term "distribute" includes making the Software Product available (either intentionally or unintentionally) to third parties for copying or using. Each time You distribute the Software Product or any original copy of the Software Product, You are responsible

Product or any original copy of the Software Product, you are responsible for the recipient expressly agree to comply with the terms and conditions of the Agreement. The recipient automatically receives the license to use, copy or distribute the Software Product subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

e. RCDevs shall have no obligation to provide any maintenance, support, upgrades or new releases of the Software Product.

4. INTELLECTUAL PROPERTY OWNERSHIP, RESERVATION OF RIGHTS. Title, copyright, ownership rights, and any other intellectual property rights in and to the Software Product, including its Documentation, and each copy thereof are and shall remain the only and absolute property of RCDevs. Except as expressly stated herein, the Agreement does not grant You any intellectual property rights in the Software Product and all rights not expressly granted are reserved by RCDevs.

5. WARRANTY DISCLAIMER.

THE SOFTWARE PRODUCT IS LICENSED FREE OF CHARGE, AND THERE IS NO WARRANTY OF ANY KIND FOR THE SOFTWARE PRODUCT.

RCDevs PROVIDE THE SOFTWARE PRODUCT "AS IS" WITH ALL FAULTS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, CUSTOM, ACCURACY OF INFORMATIONAL CONTENT, SYSTEM INTEGRATION OR NON-INFRINGEMENT ARE DISCLAIMED.

THE ENTIRE RISK AS TO THE RESULTS, QUALITY AND PERFORMANCE OF THE SOFTWARE PRODUCT IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU (AND NOT RCDevs) ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

6. LIMITATION OF LIABILITY.

YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT IN NO EVENT WILL RCDevs BE LIABLE FOR ANY DAMAGES, CLAIMS OR COSTS WHATSOEVER INCLUDING ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE, CONSEQUENTIAL OR EXEMPLARY DAMAGES, INCLUDING BUT NOT LIMITED TO, DAMAGES FOR LOSS OF USE, DATA, OR OTHER INTANGIBLE LOSSES, ARISING OUT OF, OR RELATED TO THE AGREEMENT OR TO YOUR USE OR THE INABILITY TO USE THE SOFTWARE PRODUCT OR DOCUMENTATION, EVEN IF RCDevs HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS, DAMAGES OR CLAIMS.

7. TERMINATION. The license granted hereunder is effective until terminated by RCDevs, in its sole discretion, after notification. You may terminate the Agreement at any time by uninstalling and destroying all copies of the Software Product in your possession or control.

This license will terminate automatically if you fail to comply with the terms and conditions of the Agreement above. Upon such termination, you must destroy all copies of the Software Product.

The provisions of Section 5 and 6 shall survive the termination of the Agreement.

8. APPLICABLE LAW AND GENERAL PROVISIONS. The Agreement will be governed by and construed in accordance with the Luxembourg law and submitted to the Luxembourg competent courts.

The URL-link of any open-source files and libraries relating to the Software Product is located in the file docs/licenses.txt.

If you have any questions, notices or information relating to the Agreement, please use the address and contact information included with the Software Product or via the web at <http://www.rcdevs.com/>.

I agree with RCDevs WebADM terms and conditions (Yes/No): Yes

Setup WebADM as master server or slave (secondary server in a cluster) ([m]/s)? m

Setup WebADM as a Standalone CA (1) or Subordinate CA (2) ([1]/2)?

To configure WebADM as a Subordinate CA, you need to copy your Sub-CA certificate and key as PEM format in `/opt/webadm/pki/ca/ca.crt` and `/opt/webadm/pki/ca/ca.key`

- > If option 1 is chosen, CA certificate and CA key are going to be generated at the end of the setup, and you are prompted to provide the following information regarding the creation of your PKI.

Setup WebADM as a Standalone CA (1) or Subordinate CA (2) ([1]/2)? 1

Please, provide the Issuer Name of the root certificate that is going to be created for the new Certificate Authority. If none of asked attribute is provided, a default name like 'WebADM CA# XXXXX' will be configured)

This information matter and will be visible in every certificates issued by WebADM!

Country code (e.g. LU): LU

Organization Name (e.g. RCDevs Security SA): RCDevs Support SA

Organizational Unit Name (e.g. Certificate Authority Services): RCDevs Support Root CA

Common Name (e.g. RCDevs Root CA): Support Certificate Authority

- > If option 2 is chosen, CA certificate and CA key must be already copied as `ca.crt` and `ca.key` in PEM format in `/opt/webadm/pki/ca/` folder.

You are then prompted to choose your directory type and schema setup for AD:

- 1) Default configuration (local RCDevs Directory)
 - 2) Other generic LDAP server (Novell eDirectory, Oracle, OpenLDAP)
 - 3) Active Directory with schema extension (preferred with AD)
 - 4) Active Directory without schema extension
 - 5) Active Directory schemas Mixed (Extended and Not Extended schema setup used)
- Choose a template number [1]:

1. If you choose the RCDevs directory, RCDevs-slapd component must be already configured and running.
2. Choose option 2 for Novell eDirectory, Oracle Directory, OpenLDAP or other LDAP not listed in that setup.
3. If you choose AD with Schema extension, a small extension of your AD schema will be required. The extension consists of adding few new objectClasses and attributes to your schema that will be used by WebADM and its components to store their data. e.g: Token data will be stored in webadmData attribute. The advantage of schema extension is that you avoid potentials conflicts with another application which could store his data in same attributes used by WebADM.
4. If you choose AD without Schema extension, then you need to take care that bootableDevice objectClass, bootFile, bootParameter and serial attributes are not already used by another application. If it is the case, then you can not choose that setup, and you need to go ahead with Schema extension.
5. This option is made to support extended schemas (all Directory types) and not extended schema of Active Directory simultaneously. This option is more designed for Software as a Service (SAAS) providers or to configure WebADM with multiple LDAP tenants where schema extension is not possible on the Active Directory. With that setup, WebADM will read the schema of all LDAP tenants configured and store the data according to what is available. If Schema extension is detected, then WebADM objectClass and attributes are prioritized. If another tenant (AD schema not extended) is configured as a tenant, bootableDevice objectClass, bootFile, bootParameter and serial attributes will be used. All other tenants than Active Directory must have their schema extended!

Active Directory and Schema Extension

The schema master domain controller must be set as primary LDAP server in order to perform the schema extension from graphical setup of WebADM.

Select your option, and then you are prompted to provide the first LDAP server IP/DNS name, LDAP port and encryption:

```
Please type the name/ip of the LDAP server: ad.support.rcdevs.com
Please type the port for LDAP [389]:
Checking LDAP port 389 on ad.support.rcdevs.com... Ok
Please choose the encryption ([TLS]/SSL/NONE)?TLS
```

As I choose AD setup option in that example, I'm prompted to provide the LDAP tree base of my domain.

```
Please type domain FQDN (i.e. dc=lab,dc=local) []: dc=support,dc=rcdevs,dc=com
```

Here the service account (proxy_user) distinguished name (DN) is asked:

Please type a service account with read/write access to LDAP:
CN=svc_webadm,cn=Users,dc=support,dc=rcdevs,dc=com

Enter the password of the account previously provided.

Please type the user password for CN=svc_webadm,cn=Users,DC=support,DC=rcdevs,DC=com:

WebADM test the LDAP access with the provided information and credentials:

Testing user access...Ok

Provide here the super_admin account or group DN. super_admin is/are allowed to log in to WebADM Admin portal and manage the products.

Please type an account or group DN with read/write access to LDAP (multiple accounts and groups can be configured later in webadm.conf. Nasted groups are not supported for super_admins):
cn=grp_webadm_admins,cn=Users,dc=support,dc=rcdevs,dc=com

All graphical configuration performed through WebADM Admin GUI is stored in an LDAP container or an OrganizationalUnit.
Provide the DN of the object.

Please type the WebADM container [cn=WebADM,dc=support,dc=rcdevs,dc=com]:
ou=webadms,dc=support,dc=rcdevs,dc=com

Provide then the FQDN of this server. This will be used for WebADM SSL certificate/key generation.

Enter the server fully qualified host name (FQDN) [webadm1.support.rcdevs.com]:
webadm1.support.rcdevs.com

Setup is finishing. In case you chosen standalone CA at the begining of this setup, the CA certificate and key are going to be generated. Certificate use by this WebADM node is also going to be generated and signed by the Rsignd service (CA).

```
Adding CA certificate to the local trust list... Ok
Generating SSL private key... Ok
Creating SSL certificate request... Ok
Signing SSL certificate with CA... Ok
Creating webadm system user... Ok
Setting file permissions... Ok
Adding systemd service... Ok
Adding logrotate scripts... Ok
Generating secret key string... Ok
WebADM has successfully been setup.
```

7.2. Failover configuration (Enterprise license required)

Failover is handled by Watchd service. In order to configure WebADM failover to external services (LDAP, SQL, SMTP...) you need to manually edit the `/opt/webadm/conf/servers.xml` and declare the different connectors. If you are going to create a cluster of multiple WebADM servers, then you can already configure the different Session Servers. This file is going to be copied across on the different nodes of your WebADM cluster. Configure it with IPs or DNS names in order to avoid edition of this file on slave nodes by using localhost declaration.

Found below, an example of servers.xml file configured for failover:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<Servers>
```

```
<!--
```

```
*****
```

```
*** WebADM Remote Server Connections ***
```

```
*****
```

You can configure multiple instances for each of the following servers. At login, WebADM will try to connect the configured servers in the same order they appear in this file and uses the first one it successfully establishes the connection to. If the server connection goes down, it will automatically fail over to the next configured server.

Any special characters must be encoded in XML compliant format. At least one LDAP server and one SQL server is required to run WebADM. Supported servers: OpenLDAP, Active Directory, Novell eDirectory, 389.

Allowed LDAP parameters are:

- name: server friendly name
- host: server hostname or IP address
- port: LDAP port number
default and TLS: 389
default SSL: 636

- encryption: connection type
allowed type are NONE, SSL and TLS
default: 'NONE'
 - ca_file: Trusted CA for SSL and TLS
 - cert_file: client certificate file
 - key_file: client certificate key
- >

```
<LdapServer name="AD 1"  
  host="ad1.support.rcdevs.com"  
  port="389"  
  encryption="TLS"  
  ca_file="" />
```

```
<LdapServer name="AD 2"  
  host="ad2.support.rcdevs.com"  
  port="389"  
  encryption="TLS"  
  ca_file="" />
```

```
<LdapServer name="AD 3"  
  host="ad3.support.rcdevs.com"  
  port="389"  
  encryption="TLS"  
  ca_file="" />
```

```
<LdapServer name="AD 4"  
  host="ad4.support.rcdevs.com"  
  port="389"  
  encryption="TLS"  
  ca_file="" />
```

<!--

SQL servers are used for logs; message localizations and inventories.

Supported servers: MySQL5, MySQL8, PostgreSQL, MSSQL, Sybase, Oracle, SQLite.

Allowed SQL parameters are:

- type: MySQL5, MySQL8, MariaDB, PostgreSQL, MSSQL, SQLite.
- name: server friendly name
- host: server hostname or IP address
- port: SQL port number (depends on server type)
- user: database user
- password: database password
- database: database name
- charset: character set (use latin1 if you get unicode issues)
- encryption: connection type allowed type are NONE, SSL and TLS
- ca_file Trusted CA for SSL and TLS
- cert_file: client certificate file
- key_file: client certificate key

With SQLite, only the 'database' must be set and other parameters are ignored. The database is the full path to an SQLite DB file where WebADM has full write access.

With Oracle, you can optionally use TNS names. If the 'tnsname' is set then the 'host' and 'port' parameters are ignored and a tnsnames.ora file must exist under the conf/ directory.

-->

```
<SqlServer name="SQL Server 1"
  type="MariaDB"
  host="webadm1.support.rcdevs.com"
  user="webadm"
  password="webadm"
  database="webadm"
  encryption="NONE" />
```

```
<SqlServer name="SQL Server 2"
  type="MariaDB"
  host="webadm2.support.rcdevs.com"
  user="webadm"
  password="webadm"
  database="webadm"
  encryption="NONE" />
```

<!--

A session server is required for storing/sharing persistent memory data on your WebADM server(s). You must specify two servers with clustering. The session server is based on Redis6 which is included in WebADM. With WebADM >= 2.1.5, TLS encryption is used by default on port 4000!

-->

```
<SessionServer name="Session Server 1"
  host="webadm1.support.rcdevs.com"
  port="4000"
  secret="my_secret" />
```

```
<SessionServer name="Session Server 2"
  host="webadm2.support.rcdevs.com"
  port="4000"
  secret="my_secret" />
```

<!--

A PKI server (or CA) is required for signing user certificates. The RSign PKI server is included in WebADM. So you can keep the default settings here.

-->

```
<PkiServer name="PKI Server 1"
```

```
host="webadm1.support.rcdevs.com"  
port="5000"  
secret="my_secret" />
```

```
<PkiServer name="PKI Server 2"  
  host="webadm2.support.rcdevs.com"  
  port="5000"  
  secret="my_secret" />
```

```
<!--  
HTTP proxy servers can be used by WebADM for connecting  
remote Web services and version checking.  
-->
```

```
<ProxyServer name="HTTP Proxy 1"  
  host="proxy1.support.rcdevs.com"  
  port="8080"  
  user=""  
  password=""  
  ca_file="" />
```

```
<ProxyServer name="HTTP Proxy 2"  
  host="proxy2.support.rcdevs.com"  
  port="8080"  
  user=""  
  password=""  
  ca_file="" />
```

```
<!--  
SMTP mail servers can be used by WebADM for sending emails.  
If no server is specified, WebADM will use the local mailer  
in /usb/sbin/sendmail to send emails.  
-->
```

```
<MailServer name="SMTP Server 1"  
  host="mail1.support.rcdevs.com"  
  port="25"  
  user=""  
  password=""  
  encryption="NONE"  
  ca_file="" />
```

```
<MailServer name="SMTP Server 2"  
  host="mail2.support.rcdevs.com"  
  port="25"  
  user=""  
  password=""  
  encryption="NONE"  
  ca_file="" />
```

</Servers>

See SQL master-master databases configuration in the clustering section of that documentation.

7.3. WebADM Configuration file (webadm.conf)

The configuration file we are focusing now is `/opt/webadm/conf/webadm.conf`. After the master setup, mandatory parameters has been already configured in that file in order to start WebADM services but everything is not setup. That configuration file contain the main configuration of WebADM regarding LDAP attributs, super_admins, services account, encryption key, HSM configuration and more.... All settings that can be changed are well documented in the configuration file. LDAP Attributes, LDAP objectClasses should not be changed. They have been configured according to the LDAP setup you choose during the setup script.

Found below, an example of webadm.conf file:

```
#
# WebADM Server Configuration
#
# Administrator Portal's authentication method.
# - PKI: Requires client certificate and login password.
# - UID: Requires domain name, login name and password.
# - DN: Requires login DN and password.
# - OTP: Like UID with an OTP challenge.
# - U2F: Like UID with a FIDO-U2F challenge.
# - MFA: Like UID with both OTP and FIDO-U2F challenge.
# Using certificates is the most secure login method. To use certificate login,
# you must log in WebADM and create a login certificate for your administrators.
# The UID mode requires a WebADM domain to exist and have its User Search Base
# set to the subtree where are located the administrator users. When using UID
# and if there is no domain existing in WebADM, the login mode is automatically
# forced to DN. You will also need to log in with the full user DN and set up
# a WebADM domain to be able to use the UID login mode.admin_auth UID
admin_auth UID

# Show the registered domain list when admin_auth is set to UID, OTP or U2F.
# And set a default admin login domain when auth_mode is set to these methods.
list_domains Yes
#default_domain "Default"

# Manager API's authentication method. Only UID, PKI and DN are supported here.
# If you set the admin_auth with multi-factor (PKI, OTP or U2F), then you must
# either use manager_auth PKI or UID with a list of allowed client IPs.
#manager_auth UID
#manager_clients "192.168.0.10","192.168.0.11"

# User level changes the level of feature and configuration for all applications.
# WebADM proposes three levels: Beginner, Intermediate and Expert. The default
```

```

# WebADM proposes three levels: beginner, intermediate and expert. The default
# level (Expert) is recommended as it provides access to all the RCDevs features.
#user_level Expert

# If your LDAP directory is setup with a base DN (ex. dc=mydomain,dc=com on AD),
# you can optionally set the base_treebase suffix and omit the suffix in other
# LDAP configurations like proxy_user, super_admins and containers.
ldap_treebase "dc=support,dc=rcdevs,dc=com"

# The proxy user is used by WebADM for accessing LDAP objects over which the
# admin user does not have read permissions or out of an admin session.
# The proxy user should have read permissions on the whole LDAP tree,
# and write permissions on the users/groups used by the WebApps and WebSrvs.
# The use of a proxy user is required for WebApps and WebSrvs.
# With ActiveDirectory, you can use any Domain Administrator DN as a proxy user,
# which should look like cn=Administrator,cn=Users,dc=mydomain,dc=com.
proxy_user "CN=svc_webadm,cn=Users"
proxy_password "my_password"

# Super administrators have extended WebADM privileges such as setup permissions,
# additional operations and unlimited access to any LDAP encrypted data. Access
# restriction configured in the WebADM OptionSets do not apply to super admins.
# You can set a list of individual LDAP users or LDAP groups here.
# With ActiveDirectory, your administrator account should be is something like
# cn=Administrator,cn=Users,dc=mydomain,dc=com. And you can replace the sample
# super_admins group on the second line with an existing security group.
super_admins "cn=grp_webadm_admins,cn=Users"

# LDAP objectclasses
container_oclasses "container", "organizationalUnit", "organization", "domain", "locality", "country", \
    "openldaprootdse", "treeroot"
# user_oclasses is used to build the LDAP search filter with 'Domain' auth_mode.
# If your super admin user does not have one of the following objectclasses,
# add one of its objectclasses to the list.
user_oclasses "user", "account", "person", "inetOrgPerson", "posixAccount"
group_oclasses "group", "groupOfNames", "groupOfUniqueNames", "dynamicGroup", "posixGroup"

# With ActiveDirectory 2003 only, you need to add the 'user' objectclass to the
# webadm_account_oclasses and the 'group' objectclass to the webadm_group_oclasses.
webadm_account_oclasses "webadmAccount"
webadm_group_oclasses "webadmGroup"
webadm_config_oclasses "webadmConfig"

# LDAP attributes
certificate_attrs "userCertificate"
password_attrs "userPassword", "unicodePwd", "sambaNTPassword"
uid_attrs "uid", "samAccountName", "userPrincipalName"
member_attrs "member", "uniqueMember"
memberof_attrs "memberOf", "groupMembership"
memberuid_attrs "memberUid"

```



```
language_attrs    "preferredLanguage"
mobile_attrs     "mobile"
mail_attrs       "mail"
webadm_data_attrs "webadmData"
webadm_settings_attrs "webadmSettings"
webadm_type_attrs "webadmType"
webadm_voice_attrs "webadmVoice"
```

```
# Set the LDAP container required by WebADM to store its configuration objects.
config_container "ou=webadms"
```

```
# You can alternatively configure each configuration container independently.
```

```
#domains_container "cn=Domains,cn=WebADM"
#clients_container "cn=Clients,cn=WebADM"
#devices_container "cn=Devices,cn=WebADM"
#webapps_container "cn=WebApps,cn=WebADM"
#websrvs_container "cn=WebSrvs,cn=WebADM"
#adminroles_container "cn=AdminRoles,cn=WebADM"
#optionsets_container "cn=OptionSets,cn=WebADM"
#mountpoints_container "cn=MountPoints,cn=WebADM"
```

```
# You can set here the timeout (in seconds) of a WebADM session.
```

```
# Web sessions will be closed after this period of inactivity.
```

```
# The Manager Interface cookie-based sessions are disabled by default.
```

```
# admin_session and manager_session can be set in the form 'shared:900'
```

```
# in order to force sessions to be stored in the Session Servers instead of SHM.
```

```
admin_session 3600
manager_session 0
webapps_session 600
```

```
# You can set here the WebADM internal cache timeout. A normal value is one hour.
```

```
cache_timeout 3600
```

```
# Application languages
```

```
languages "EN","FR","DE","HU","ES","IT","FI","JP"
```

```
# WebADM encrypts LDAP user data, sensitive configurations and user sessions with
```

```
# AES-256. The encryption key(s) must be 256bit base64-encoded random binary data.
```

```
# Use the command 'openssl rand -base64 32' to generate a new encryption key.
```

```
# Warning: If you change the encryption key, any encrypted data will become invalid!
```

```
# You can set several encryption keys for key rollout. All the defined keys are used
```

```
# for decrypting data. And the first defined key is used to (re-)encrypt data.
```

```
# Two encryption modes are supported:
```

```
# Standard: AES-256-CBC (default)
```

```
# Advanced: AES-256-CBC with per-object encryption (stronger)
```

```
encrypt_data yes
encrypt_mode Standard
encrvpt hsm No
```

```
encrypt_key "cq19TEHgHLQuO09DXzjOw30rrQDLsPkt3NiL6I3BH2w="
```

```
# Hardware Cryptographic Module
```

```
#hsm_driver "/usr/local/lib/libsoftsm2.so"
```

```
#hsm_slot 274906134
```

```
#hsm_key "TestKey"
```

```
#hsm_pin 12345678
```

```
# The data store defines which back-end is used for storing user data and settings.
```

```
# By default WebADM stores any user and group metadata in the LDAP objects. By setting
```

```
# the data_store to SQL, these metadata are stored in a dedicated SQL table.
```

```
# LDAP remains the preferred option because it maximizes the system consistency.
```

```
# SQL should be used only if you need read-only LDAP access for the proxy_user.
```

```
data_store LDAP
```

```
# The record store defines which back-end is used to store SpanKey records.
```

```
# Choose SQL to store records in the database and NAS to store on a shared NAS folder.
```

```
# With NAS, the store_path must be configured and accessible from all cluster nodes.
```

```
record_store SQL
```

```
#record_path "/mnt/records"
```

```
# The group mode defines how WebADM will handle LDAP groups.
```

```
# - Direct mode: WebADM finds user groups using the memberof_attrs defined above.
```

```
# In this case, the group membership is defined in the LDAP user objects.
```

```
# - Indirect mode: WebADM finds user groups by searching group objects which contain
```

```
# the user DN as part of the member_attrs.
```

```
# - Auto: Both direct and indirect groups are used.
```

```
# - Disabled: All LDAP group features are disabled in WebADM.
```

```
# By default (when group_mode is not specified) WebADM handles both group modes.
```

```
group_mode Auto
```

```
# LDAP cache increases a lot of performances under high server loads. The cache limits
```

```
# the number of LDAP requests by storing resolved user DN and group settings. When
```

```
# enabled, results are cached for 300 secs.
```

```
ldap_cache Yes
```

```
# LDAP routing enables LDAP request load-balancing when multiple LDAP servers are
```

```
# configured in servers.xml. You should enable this feature only if the LDAP server
```

```
# load becomes a bottleneck due to a big amount of users (ex. more than 10000 users).
```

```
#ldap_routing No
```

```
# You can optionally disable some features if you run multiple WebADM servers with
```

```
# different purposes. For example, if you don't want to provide admin portal on an
```

```
# Internet-exposed WebApps and WebSrvs server.
```

```
# By default, all the functionalities are enabled.
```

```
enable_admin Yes
```

```
enable_manager Yes
```

```
enable_webapps Yes
```

enable_websrvs Yes

Enable syslog reporting (disabled by default). When enable, system logs are sent
to both the WebADM log files and syslog.

#log_debug No

#log_mixsql No

#log_syslog No

#syslog_facility LOG_USER

#syslog_format CEF

Alerts are always recorded to the SQL Alert log. Additionally, when alert_email
or alert_mobile is defined, the alerts are also sent by email/SMS.

alert_email "alert@support.rcdevs.com"

alert_mobile "+33 12345678"

Protect WebADM against bruteforce attacks on the WebApps by blacklisting source IPs
for 20 seconds after 5 failed login attempts.

ip_blacklist Yes

You can publish WebADM applications and OpenOTP mobile endpoint over Internet using
a reverse proxy (WAF) or RCDevs WebADM Publishing Server (WAProxy).
Set the IP address(es) of your reverse-proxy or WAProxy server(s). WebADM expects
the HTTP_X_FORWARDED_FOR and HTTP_X_FORWARDED_HOST headers from reverse proxies!
Use 'waproxy_proxies' ONLY if you are using RCDevs WAProxy as reverse-proxy!

reverse_proxies "192.168.0.100", "192.168.0.101"

waproxy_proxies "192.168.0.102", "192.168.0.104"

The 'public_hostname' is mandatory to let WebADM know your public endpoints' URLs.

Use the public DNS name of your reverse proxy or WAProxy server without a scheme.

The setting used to be named 'waproxy_pubaddr' in WebADM versions before v2.3.12.

public_hostname "otp.support.rcdevs.com"

Check for new product versions and license updates on RCDevs' website.

These features require outbound Internet access from the server.

cloud_services yes

WebApps theme (default or flat)

Comment the following line to disable the default theme.

webapps_theme "default"

End-user message templates

The following variables are available: %USERNAME%, %USERDN%, %USERID%, %DOMAIN%,
%APPNAME%

Additional variables are available depending on the context: %APPNAME%, %APPID%, %TIMEOUT%,
%EXPIRES%

app_unlock_subject "Unlocked access to %APPNAME%"

app_unlock_message "Hello %USERNAME%,\r\n\r\nYou have a one-time access to the
%APPNAME%.\r\n\r\nYour access will automatically expire %EXPIRES%."

ldap_expire_subject "Login password near expiration"

ldap_expire_message "Hello %USERNAME%,\r\n\r\nYour login password will expire %EXPIRES%.\r\n\r\nPlease
reset your password before expiration!\r\n\r\nRegards"

```

cert_expire_subject "Login certificate near expiration"
cert_expire_message "Hello %USERNAME%,\r\n\r\nYour login certificate will expire %EXPIRES%.\r\nPlease
renew your certificate before expiration!\r\n\r\nRegards"
access_sign_subject "Agreement signature required for %CLIENT%"
access_sign_message "Hello %USERNAME%,\r\n\r\nPlease sign the agreement in order to access
%CLIENT%.\r\nThe signature request expire %EXPIRES%."
no_badgeout_subject "Forgot badge-out %EXPIRES%"
no_badgeout_message "Hello %USERNAME%,\r\n\r\nYou did not badge-out since %EXPIRES%.\r\nPlease
do not forget to badge out today!\r\n\r\nRegards"
no_badgein_subject "Badging required for %CLIENT%"
no_badgein_message "Hello %USERNAME%,\r\n\r\nYou tried to login to %CLIENT% without
badging.\r\nPlease badge-in and retry!\r\n\r\nRegards"

# Personalization options
# You can customize your organization name, logo file and website URL.
# The logo file must be PNG image with size 100x50 pixels.
org_name "RCDevs Support"
org_logo "rcdevs.png"
org_site "https://www.rcdevs.com/"
org_from "noreply@support.rcdevs.com"

# Misc options
#treeview_width 300
#treeview_items 3000
#default_portal Admin
#ldap_uidcase No
ntp_server "ad1.support.rcdevs.com"

```

Once you configured all settings you need, it is time to start WebADM services.

7.4. Rsignd configuration

Rsignd is the PKI service running with WebADM. The configuration of Rsignd is located in `/opt/webadm/conf/rsignd.conf` on all nodes.

The `rsignd.conf` file must be configured in all WebADM servers in a Master/Master installation, and the secret must match what is configured in `servers.xml` in PKI section.

```

# Log file
logfile /opt/webadm/logs/rsignd.log
pidfile /opt/webadm/temp/rsignd.pid

# Default validity period for new certificates (in days)
# The CSR signing requests may set the validity period.
user_cert_validity 365
client_cert_validity 1825
server_cert_validity 3650

```

```
# Certificate and key used for the SSL listener
rsignd_cert /opt/webadm/pki/webadm.crt
rsignd_key /opt/webadm/pki/webadm.key

# Path CA certificate files and serial
ca_cert /opt/webadm/pki/ca/ca.crt
ca_key /opt/webadm/pki/ca/ca.key
ca_serial /opt/webadm/pki/ca/serial

# Serial number format (hex or dec)
serial_format hex

# Set to yes if the CA or rsignd private keys requires a decryption password.
# PEM passwords will be prompted at WebADM startup.
ca_password no
rsignd_password no

# HSM certificate authority (CA)
# The HSM model and PIN code are configured in webadm.conf.
hsm_ca no
hsm_keyid 0

#
# Directory or file containing trusted CA certificates (in PEM format)
# After adding a new certificate, type a "make" in the "trusted_ca_path"
# to rebuild certificate's hash.
# This is needed for rsignd to read the trusted CA certificates.
# Comment "trusted_path" to disable rsignd certificate's trust restrictions.
trusted_path /opt/webadm/pki/trusted

#
# Client sections
#
# Declare here the Rsign clients with IP addresses or hostnames.
# In cluster mode, the client WebADM server(s) must be defined here!

client {
    hostname webadm1.support.rcdevs.com
    secret my_secret
}

client {
    hostname webadm2.support.rcdevs.com
    secret my_secret
}
```

Here, I configured `webadm2.support.rcdevs.com` as Rsignd client which is the WebADM slave node that I will configure

later in this documentation. The master is declared by default during the master setup.

Alternatively, the CA Key can be stored on an HSM and the HSM can be involved for each certificate signing request. You have to configure first your HSM with WebADM in `webadm.conf` then program the HSM with your CA key (contained in `/opt/webadm/pki/ca/` folder) and provide the `hsm_keyid` value in `rsigned.conf`. Then set `hsm_ca` setting to `yes`. WebADM support [MirKey HSM](#), [SmartCard HSM](#) and PKCS11 standard HSMs.

8. Set up the SQL Database

WebADM uses an SQL database to store audit logs, localized messages, SSL certificates, statistics, inventoried hardware tokens. Application configurations, users and their metadata are directly stored in LDAP rather than in the databases. WebADM supports both MySQL and PostgreSQL databases. Other databases are not currently fully tested. You must create a webadm database on your SQL server and a webadm user with password webadm, having full permissions on that database. Edit the `/opt/webadm/conf/servers.xml` file and adjust the SQL Server parameters such as the database username and password. Note that the graphical setup process will create the required tables.

8.1. Installation of MariaDB server

Install with Debian repository:

```
root@webadm:~# apt-get install mariadb-server
```

Install with yum repository:

```
root@webadm:~# yum install mariadb-server
```

Then enable, start mariadb service, and execute built-in script to secure installation:

```
root@webadm:~# systemctl enable mariadb
root@webadm:~# systemctl start mariadb
root@webadm:~# mysql_secure_installation
```

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current password for the root user. If you've just installed MariaDB, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] Y

New password:

Re-enter new password:

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? [Y/n] Y

... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y

... Success!

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] Y

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] Y

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

8.2. Create the WebADM database

Finally, You can use this script to create the `database`, `user`, and `grant` the privileges on a specific database to a user in the cluster:

```
[root@webadm1 ~]# cat /opt/webadm/doc/scripts/create_mysqlldb

#!/bin/bash
# This SQL script initializes the WebADM database on a MySQL Server

# print the usage of the command
usage() {
    echo "Usage: $0 [-s] [-h SQL_IP_OR_DNS] [-P SQL_PORT] -d DATABASE_NAME -u WEBADM_USERNAME -p
WEBADM_PASSWORD -w WEBADM_IP1[,WEBADM_IP2] [-c CHARACTER_SET -o COLLATION] [-n]" 1>&2
    echo "-s use socket for connection (disables -h and -P)" 1>&2
    echo "-h IP or DNS of the SQL server (default to localhost)" 1>&2
    echo "-P port of the SQL server (default to 3306)" 1>&2
    echo "-d name of the webadm database" 1>&2
    echo "-u webadm username" 1>&2
    echo "-p webadm password" 1>&2
    echo "-w List of IPs of WebADM servers, comma separated" 1>&2
    echo "-c character set for database (default to latin1)"
    echo "-o collation for database (default to latin1_swedish_ci)"
    echo "-n no password mode"
}

# print the usage and exit the program
exit_abnormal() {
    usage
    exit 1
}

check_mysql_authentication() {
    if [ "$#" -eq 0 ]; then
        mysql -uroot -e "" >/dev/null 2>&1
    elif [ "$#" -eq 1 ]; then
        mysql -uroot -p"$1" -e "" >/dev/null 2>&1
    elif [ "$#" -eq 2 ]; then
        mysql -uroot -h "$1" -P "$2" -e "" >/dev/null 2>&1
    elif [ "$#" -eq 3 ]; then
        mysql -uroot -p"$1" -h "$2" -P "$3" -e "" >/dev/null 2>&1
    fi
}

execute_sql_script() {
    if [ "$#" -eq 0 ]; then
        mysql -uroot -e "$SQL_SCRIPT" >/tmp/create_mysqlldb.log 2>&1
    elif [ "$#" -eq 1 ]; then
        mysql -uroot -p"$1" -e "$SQL_SCRIPT" >/tmp/create_mysqlldb.log 2>&1
    fi
}
```



```

mysql -uroot -p $1 -e "$SQL_SCRIPT" >/tmp/create_mysqlldb.log 2>&1
elif [ "$#" -eq 2 ]; then
    mysql -uroot -h "$1" -P "$2" -e "$SQL_SCRIPT" >/tmp/create_mysqlldb.log 2>&1
elif [ "$#" -eq 3 ]; then
    mysql -uroot -p"$1" -h "$2" -P "$3" -e "$SQL_SCRIPT" >/tmp/create_mysqlldb.log 2>&1
fi
}

command -v mysql >/dev/null 2>&1 || { echo >&2 "mysql command is required but it's not installed.
Aborting."; exit 1; }

# Get the values of the different command arguments
while getopts "snd:h:u:p:w:P:c:o:" options; do
    case "${options}" in
        s)
            USE_SOCKET=1
            ;;
        n)
            NO_PASSWORD=1
            ;;
        h)
            SQL_IP_OR_DNS=${OPTARG}
            ;;
        P)
            SQL_PORT=${OPTARG}
            ;;
        d)
            DATABASE_NAME=${OPTARG}
            ;;
        u)
            WEBADM_USERNAME=${OPTARG}
            ;;
        p)
            WEBADM_PASSWORD=${OPTARG}
            ;;
        w)
            mapfile -d "," -t WEBADM_IPS <<(echo -n "${OPTARG}")
            ;;
        c)
            CHARACTER_SET=${OPTARG}
            ;;
        o)
            COLLATION=${OPTARG}
            ;;
        *)
            exit_abnormal
            ;;
    esac
done

```

```
USE_SOCKET=${USE_SOCKET:-0}

if [ -z "${SQL_IP_OR_DNS}" ]; then
    SQL_IP_OR_DNS="localhost"
fi

if [ -z "${SQL_PORT}" ]; then
    SQL_PORT="3306"
fi

if [ -z "${NO_PASSWORD}" ]; then
    NO_PASSWORD=0
fi

if [ -z "${DATABASE_NAME}" ]; then
    usage
    exit
fi

if [ -z "${WEBADM_USERNAME}" ]; then
    usage
    exit
fi

if [ -z "${WEBADM_PASSWORD}" ]; then
    usage
    exit
fi

if [ -z "${WEBADM_IPS[*]}" ]; then
    usage
    exit
fi

if [ -z "${CHARACTER_SET}" ]; then
    CHARACTER_SET="latin1"
fi

if [ -z "${COLLATION}" ]; then
    COLLATION="latin1_swedish_ci"
fi

if [ "$NO_PASSWORD" -eq 0 ]; then
    echo -n "Please enter password of SQL root username: "
    read -r -s PASSWORD_ROOT
    echo
fi

if [ "$USE_SOCKET" -eq 1 ]; then
    if [ "$NO_PASSWORD" -eq 1 ]; then
```

```

if [ "$NO_PASSWORD" -eq 1 ]; then
    check_mysql_authentication
else
    check_mysql_authentication "$PASSWORD_ROOT"
fi
else
if [ "$NO_PASSWORD" -eq 1 ]; then
    check_mysql_authentication $SQL_IP_OR_DNS $SQL_PORT
else
    check_mysql_authentication "$PASSWORD_ROOT" $SQL_IP_OR_DNS $SQL_PORT
fi
fi

if [ $? == 1 ]; then
    echo "Authentication error. Please check credentials. Exiting!"
    exit 1
fi

# Creation of users and then grant (split in order to be compatible with MySQL version 8 which does not
allow creation of users during GRANT operation)
SQL_SCRIPT="CREATE DATABASE IF NOT EXISTS $DATABASE_NAME CHARACTER SET $CHARACTER_SET
COLLATE $COLLATION;"
SQL_SCRIPT="${SQL_SCRIPT}CREATE USER IF NOT EXISTS '$WEBADM_USERNAME'@'localhost'
IDENTIFIED BY '$WEBADM_PASSWORD';"
SQL_SCRIPT="${SQL_SCRIPT}GRANT ALL PRIVILEGES ON $DATABASE_NAME.* TO
'$WEBADM_USERNAME'@'localhost';"
for i in "${!WEBADM_IPS[@]}"; do
    SQL_SCRIPT="${SQL_SCRIPT}CREATE USER IF NOT EXISTS
'$WEBADM_USERNAME'@'${WEBADM_IPS[$i]}' IDENTIFIED BY '$WEBADM_PASSWORD';"
    SQL_SCRIPT="${SQL_SCRIPT}GRANT ALL PRIVILEGES ON $DATABASE_NAME.* TO
'$WEBADM_USERNAME'@'${WEBADM_IPS[$i]}';"
done
SQL_SCRIPT="${SQL_SCRIPT}FLUSH PRIVILEGES;"

if [ "$USE_SOCKET" -eq 1 ]; then
    if [ "$NO_PASSWORD" -eq 1 ]; then
        execute_sql_script
    else
        execute_sql_script "$PASSWORD_ROOT"
    fi
else
    if [ "$NO_PASSWORD" -eq 1 ]; then
        execute_sql_script $SQL_IP_OR_DNS $SQL_PORT
    else
        execute_sql_script "$PASSWORD_ROOT" $SQL_IP_OR_DNS $SQL_PORT
    fi
fi

if [ $? == 1 ]; then
    cat /tmp/create_mysqlldb.log

```

```
rm -f /tmp/create_mysqlldb.log
else
  echo "Creation of database and users was a success!"
fi
```

Example:

```
cd /opt/webadm/doc/scripts/

[root@webadm1 scripts]# ./create_mysqlldb -d webadm -u webadm -n -w
webadm2.support.rcdevs.com,ad1.support.rcdevs.com -p "my_password"
Creation of database and users was a success!
```

Once the DB and user are created, start WebADM services and login on the WebADM Admin portal to finish the graphical setup and to create the SQL tables.

9. Start WebADM services and Run the Graphical Setup

Start WebADM services with the following command:

```
[root@webadm1 ~]# /opt/webadm/bin/webadm start
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Ok

Found Trial license (RCDEVSSUPPORT)
Licensed by RCDevs Security SA to RCDevs Support
Licensed product(s): OpenOTP,SpanKey

Starting WebADM PKI service... Ok
Starting WebADM Session service... Ok
Starting WebADM Watchd service... Ok
Starting WebADM HTTP service... Ok

Checking server connections...
Connected LDAP server: LDAP Server (ad1.support.rcdevs.com)
Connected SQL server: SQL Server (webadm1.support.rcdevs.com)
Connected PKI server: PKI Server (webadm1.support.rcdevs.com)
Connected Session server: Session Server (webadm1.support.rcdevs.com)
Connected HTTP Proxy: Proxy Server (proxy1.support.rcdevs.com)
Connected SMTP Proxy: SMTP Server (mail1.support.rcdevs.com)

Checking LDAP proxy user access... Ok
Checking SQL database access... Ok
Checking PKI service access... Ok
Checking Mail service access... Ok
Checking HTTP Proxy access... Ok
Checking Cloud service access... Ok
```

Enter WebADM with your `super_admin` account and run the graphical setup. The login URL is

<https://<your-server-address>>. Only a `super_admins` and other `_admins` configured in `webadm.conf` or through graphical configuration can access this interface.

Important

Until the graphical setup is done and at least the first WebADM Domain is created, you must log in with the LDAP DN of your `super_admins` and not the username. When the graphical setup will be completed and at least one WebADM Domain is created for the LDAP tree where your administrator is stored, you will be able to log in with username and password (UID login mode).

Important

If you use RCDevs Directory Server, the admin DN is `cn=admin,o=root`. The default password is `password`.

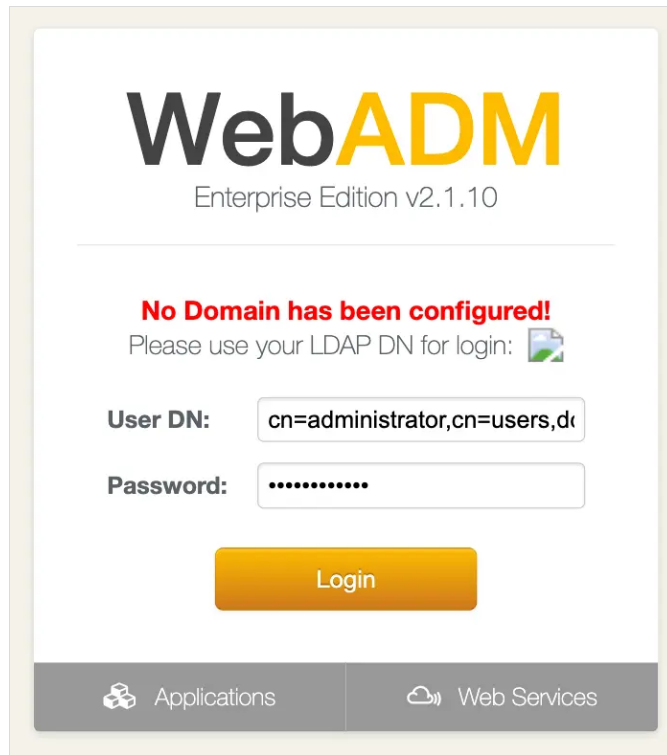


Figure 3. WebADM Admin Portal Login (Active Directory Server)

WebADM requires DN-based login until the setup is completed. Then it will use the login mode as configured in the `/opt/webadm/conf/webadm.conf` file.

The Setup wizard will appear on the home page when you enter the WebADM Admin Portal.

WebADM Enterprise Edition v2.1.10
Copyright © 2010-2022 RCDevs Security, All Rights Reserved

Home Admin Create Search Import Databases Statistics Applications About Logout

WebADM Setup

Your WebADM installation is not completely configured!
Please run the following setup actions to finish configuring WebADM.

Checking LDAP schema
Reading schema objectclasses... **Ok**
Reading schema attributes... **Ok**
Checking account objectclass... **Ok**
Checking group objectclass... **Ok**
Checking config objectclass... **Ok**
Checking data attribute... **Ok**
Checking settings attribute... **Ok**
Checking type attribute... **Ok**
Checking voice attribute... **Ok**

Checking SQL database
Checking database connection... **Ok**
Reading database tables... **Missing**

Create/Update SQL database tables

Checking WebADM proxy user
Checking proxy user exists... **Ok**
Checking proxy user bind... **Ok**

Checking WebADM super admins
Checking super admin 'cn=administrator'... **Ok**
Checking super admin 'cn=Domain Admins'... **Ok**

Checking LDAP permissions
Tree root: *DC=support,DC=rcdevs,DC=com* (Microsoft)
Checking proxy user permissions... **Ok**

Checking default LDAP objects
Checking domains container... **Missing**
Checking clients container... **Missing**
Checking devices container... **Missing**
Checking webapps container... **Missing**
Checking websrvs container... **Missing**
Checking optionsets container... **Missing**
Checking adminroles container... **Missing**
Checking mountpoints container... **Missing**

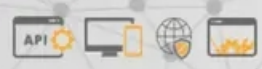
Create default containers and objects

You must logout when setup is completed.

WebADM will run very slow and will not be functional until the graphical setup has been completed. The MountPoints, OptionSets, WebSrvs, WebApps and many features are kept disabled until the setup is completed.

The graphical setup process will:

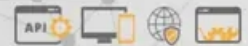
- > Create the required database tables;
- > Register the required LDAP schema object classes and attributes (with Novell eDirectory and Microsoft ActiveDirectory), if schema extension is not available;
- > Create the proxy user (if not already existing);
- > Set up the proxy user permissions (on Novell eDirectory);
- > Create the WebADM LDAP containers (as defined in the `/opt/webadm/conf/webadm.conf` file).



Database Setup

Creating table 'Admin'... Success
Creating index 'Admin_DN_idx'... Success
Creating table 'Manag'... Success
Creating index 'Manag_DN_idx'... Success
Creating table 'WebApp'... Success
Creating index 'WebApp_DN_idx'... Success
Creating table 'WebSrv'... Success
Creating index 'WebSrv_DN_idx'... Success
Creating table 'Alert'... Success
Creating table 'Message'... Success
Creating table 'Inventory'... Success
Creating index 'Inventory_DN_idx'... Success
Creating table 'Record'... Success
Creating index 'Record_DN_idx'... Success
Creating index 'Record_Session_idx'... Success
Creating table 'Certificate'... Success
Creating table 'Statistic'... Success

Ok



- Checking field Record.Stop... **Ok**
- Checking field Record.DN... **Ok**
- Checking field Record.Source... **Ok**
- Checking field Record.Host... **Ok**
- Checking field Record.Session... **Ok**
- Checking field Record.Type... **Ok**
- Checking field Record.Size... **Ok**
- Checking field Record.Data... **Ok**
- Checking field Record.Crypt... **Ok**
- Checking field Record.Store... **Ok**
- Checking table Certificate... **Ok**
- Checking field Certificate.Type... **Ok**
- Checking field Certificate.Reference... **Ok**
- Checking field Certificate.Description... **Ok**
- Checking field Certificate.Application... **Ok**
- Checking field Certificate.Start... **Ok**
- Checking field Certificate.Stop... **Ok**
- Checking field Certificate.Time... **Ok**
- Checking field Certificate.Host... **Ok**
- Checking field Certificate.Data... **Ok**
- Checking field Certificate.Active... **Ok**
- Checking field Certificate.Renew... **Ok**
- Checking table Statistic... **Ok**
- Checking field Statistic.Type... **Ok**
- Checking field Statistic.Time... **Ok**
- Checking field Statistic.Server... **Ok**
- Checking field Statistic.Group... **Ok**
- Checking field Statistic.Count... **Ok**
- Checking field Statistic.Delay... **Ok**
- Checking field Statistic.Min... **Ok**
- Checking field Statistic.Max... **Ok**

Checking WebADM proxy user

- Checking proxy user exists... **Ok**
- Checking proxy user bind... **Ok**

Checking WebADM super admins

- Checking super admin 'cn=administrator'... **Ok**
- Checking super admin 'cn=Domain Admins'... **Ok**

Checking LDAP permissions

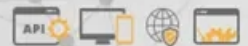
- Tree root: DC=support,DC=rcdevs,DC=com (Microsoft)
- Checking proxy user permissions... **Ok**

Checking default LDAP objects

- Checking domains container... **Missing**
- Checking clients container... **Missing**
- Checking devices container... **Missing**
- Checking webapps container... **Missing**
- Checking websrvs container... **Missing**
- Checking optionsets container... **Missing**
- Checking adminroles container... **Missing**
- Checking mountpoints container... **Missing**

Create default containers and objects

You must logout when setup is completed.



Containers Setup

- Creating WebADM Domains container cn=Domains,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**
- Creating Domain for cn=users,dc=support,dc=rcdevs,dc=com... **Success**
- Creating WebADM OptionSets container cn=OptionSets,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**
- Creating OptionSet for cn=users,dc=support,dc=rcdevs,dc=com... **Success**
- Creating WebADM AdminRoles container cn=AdminRoles,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**
- Creating WebADM WebApps container cn=WebApps,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**
- Creating WebADM WebSrvs container cn=WebSrvs,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**
- Creating WebADM Clients container cn=Clients,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**
- Creating WebADM Devices container cn=Devices,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**
- Creating WebADM MountPoints container cn=MountPoints,cn=webadm,dc=support,dc=rcdevs,dc=com... **Success**

Ok

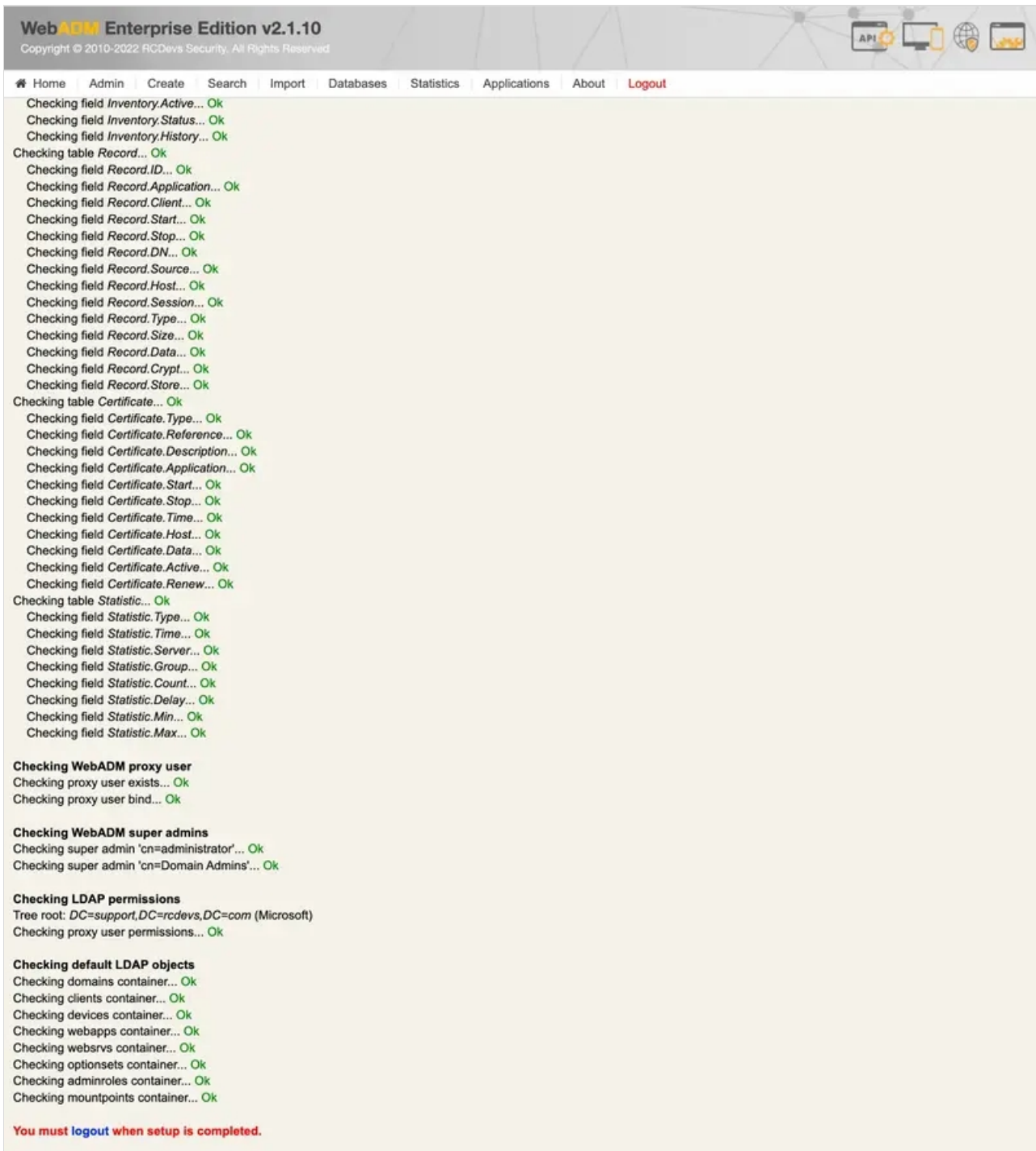
If WebADM fails to automatically create the LDAP containers, it is probably because the connected user on WebADM admin portal

doesn't have permissions to write on the container/OU configured during the setup.

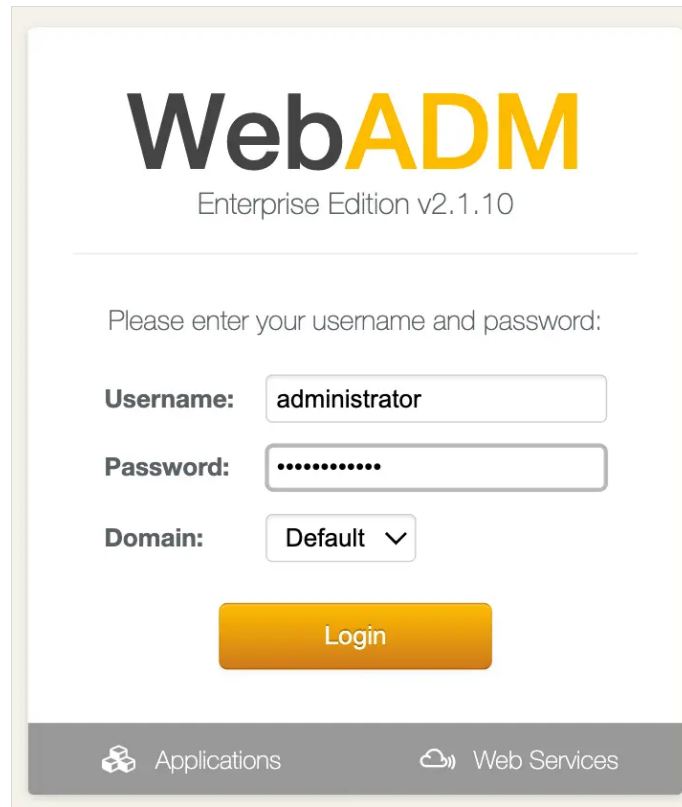
If it fails to create the SQL tables, then it is probably because the SQL user do no has the permission on the database.

If it fails to extend the schema for AD, it is probably because of the following reason:

- > The first LDAP server configured in servers.xml do not have the schema master role,
- > The user connected to WebADM do not have the schema admin and enterprise admin permissions.



Click Logout button. You can now log in with Username and password of you super_admin.



The image shows the WebADM login interface. At the top, the logo 'WebADM' is displayed in black and yellow, with 'Enterprise Edition v2.1.10' below it. A horizontal line separates the header from the login form. The form prompts the user to enter their username and password. The 'Username' field contains 'administrator', the 'Password' field is masked with dots, and the 'Domain' dropdown is set to 'Default'. A yellow 'Login' button is positioned below the fields. At the bottom of the form, there are two links: 'Applications' with a server icon and 'Web Services' with a cloud icon.

10. Advanced SQL configuration (Optional)

10.1. Configuration of Character Set

10.1.1. Configuring Charset Parameter In servers.xml

charset parameter can be used to configure what character encoding must be used when WebADM communicates with SQL servers. The setting is name `charset` in SQL section. Found below, an example of SQL declaration in servers.xml:

```
<SqlServer name="SQL Server 2"
  type="MariaDB"
  host="webadm2.support.rcdevs.com"
  user="webadm"
  password="webadm"
  charset="latin2"
  database="webadm"
  encryption="NONE" />
```

10.1.2. Checking Character Set

Note

This is important to have the same character encoding for the database and its tables. If you detect differences using next commands, please go align the configuration and change character set.

In MariaDB, the default character set is `latin1`, and the default collation is `latin1_swedish_ci`. However, this may be different in some Linux distributions. For example, the default character set is `utf8mb4` and the default collation is `utf8mb4_general_ci` in MariaDB on Debian/Ubuntu.

Run the following command `SHOW VARIABLES LIKE 'char%';` to get an overview of the default character sets:

```
-bash-4.2# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 1041
Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> SHOW VARIABLES LIKE 'char%';
```

```
+-----+-----+
| Variable_name      | Value          |
+-----+-----+
| character_set_client | latin1         |
| character_set_connection | latin1        |
| character_set_database | latin1         |
| character_set_filesystem | binary        |
| character_set_results | latin1         |
| character_set_server  | latin1         |
| character_set_system  | utf8          |
| character_sets_dir    | /usr/share/mysql/charsets/ |
+-----+-----+
8 rows in set (0.00 sec)
```

```
MariaDB [(none)]>
```

Let's view the default character set of the `webadm` database.

```
MariaDB [(none)]> SELECT default_character_set_name FROM information_schema.SCHEMATA S WHERE
schema_name = "webadm";
```

```
+-----+
| default_character_set_name |
+-----+
| latin1                    |
+-----+
1 row in set (0.00 sec)
```

```
MariaDB [(none)]>
```

Finally, check the default character set of the webadm tables and columns.

```
MariaDB [(none)]> USE webadm;
```

```
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
MariaDB [webadm]> SHOW TABLES;
```

```
+-----+
| Tables_in_webadm |
+-----+
| Admin            |
| Alert           |
| Certificate      |
| Inventory       |
| Manag           |
| Message         |
| Metadata        |
| Record         |
| Statistic       |
| WebApp          |
| WebSrv          |
+-----+
10 rows in set (0.00 sec)
```

```
MariaDB [webadm]> SELECT CCSA.character_set_name FROM information_schema.`TABLES` T,
information_schema.`COLLATION_CHARACTER_SET_APPLICABILITY` CCSA WHERE CCSA.collation_name =
T.table_collation AND T.table_schema = "webadm" AND T.table_name = "Admin";
```

```
+-----+
| character_set_name |
+-----+
| latin1             |
+-----+
1 row in set (0.01 sec)
```

```
MariaDB [webadm]> SHOW FULL COLUMNS FROM Admin;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| Field | Type      | Collation      | Null | Key | Default | Extra      | Privileges      | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID    | bigint(20) | NULL          | NO  | PRI | NULL    | auto_increment | select,insert,update,references |
| Time  | datetime   | NULL          | YES |     | NULL    |              | select,insert,update,references |
| DN    | varchar(255) | latin1_swedish_ci | YES |     | NULL    |              | select,insert,update,references |
| Source | varchar(64) | latin1_swedish_ci | YES |     | NULL    |              | select,insert,update,references |
| Session | varchar(64) | latin1_swedish_ci | YES |     | NULL    |              | select,insert,update,references |
```

```
| Text | text | latin1_swedish_ci | YES | | NULL | | select,insert,update,references |
|
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MariaDB [webadm]>
```

10.1.3. Change Character Set

Under CentOS, location of the configuration files are `/etc/my.cnf.d/server.cnf` and `/etc/my.cnf.d/client.cnf`.

For Debian/Ubuntu, location of the configuration files are `/etc/mysql/mariadb.conf.d/50-server.cnf` and `/etc/mysql/mariadb.conf.d/50-client.cnf`.

In current section, as an example, we use latin1 as character set, and latin1_swedish_ci as default collation. Please find here the link [MariaDB Supported Character Sets and Collations](#) to see other supported character and collations sets for MariaDB.

First, we add/change the default character and collation, then restart the SQL server:

```
-bash-4.2# vi /etc/my.cnf.d/server.cnf
...
# this is only for the mysqld standalone daemon
[mysqld]
character-set-server = latin1
collation-server    = latin1_swedish_ci
...
```

```
-bash-4.2# vi /etc/my.cnf.d/client.cnf
...
[client-mariadb]
default-character-set = latin1
```

Restart Mariadb services for changes takes effect:

```

-bash-4.2# systemctl restart mariadb
-bash-4.2# systemctl status mariadb -l
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2019-05-16 14:11:04 CEST; 8s ago
   Process: 14904 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited,
status=0/SUCCESS)
   Process: 14872 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
   Main PID: 14903 (mysqld_safe)
   CGroup: /system.slice/mariadb.service
           └─14903 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
             └─15089 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-
dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.log --pid-
file=/var/run/mariadb/mariadb.pid --socket=/var/lib/mysql/mysql.sock

May 16 14:11:02 rcvm7.local systemd[1]: Starting MariaDB database server...
May 16 14:11:02 rcvm7.local mariadb-prepare-db-dir[14872]: Database MariaDB is probably initialized in
/var/lib/mysql already, nothing is done.
May 16 14:11:02 rcvm7.local mariadb-prepare-db-dir[14872]: If this is not the case, make sure the
/var/lib/mysql is empty before running mariadb-prepare-db-dir.
May 16 14:11:02 rcvm7.local mysqld_safe[14903]: 190516 14:11:02 mysqld_safe Logging to
'/var/log/mariadb/mariadb.log'.
May 16 14:11:02 rcvm7.local mysqld_safe[14903]: 190516 14:11:02 mysqld_safe Starting mysqld
daemon with databases from /var/lib/mysql
May 16 14:11:04 rcvm7.local systemd[1]: Started MariaDB database server.

```

Finally, change the character and collation setting.

```

MariaDB [(none)]> ALTER DATABASE webadm CHARACTER SET latin1 COLLATE latin1_swedish_ci;
Query OK, 1 row affected (0.02 sec)

```

```

MariaDB [(none)]> USE webadm;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

```

Database changed

```

MariaDB [webadm]> SHOW TABLES;

```

```

+-----+
| Tables_in_webadm |
+-----+
| Admin          |
| Alert          |
| Certificate     |
| Inventory      |
| Manag          |
| Message        |
| Record         |

```

```
| Record |
| Statistic |
| WebApp |
| WebSrv |
+-----+
11 rows in set (0.00 sec)
```

```
MariaDB [webadm]> ALTER TABLE Admin CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE Alert CONVERT TO CHARACTER SET latin1 COLLATE latin1_swedish_ci;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE Certificate CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE Inventory CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE Manag CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE Message CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE Record CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE Statistic CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
MariaDB [webadm]> ALTER TABLE WebApp CONVERT TO CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```



```
MariaDB [webadm]> ALTER TABLE WebSrv CONVERT TO CHARACTER SET latin1 COLLATE latin1_swedish_ci;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

11. Check System Clock and Timezone

WebADM requires an accurate system clock and timezone. Your Linux server should be configured with NTP time synchronization. On RedHat/CentOS, you need to install and run the ntpd service at boot time. After installing ntpd, you can check the server time with the ntpdate command.

WebADM before version 1.5.6 required the time zone to be configured in webadm.conf. With later versions, WebADM uses the time zone which is configured at the system level. On most Linux systems, the timezone is configured by adjusting the `/etc/localtime` file or symlink.

Please, refer to the [NTP documentation](#) to achieve this.

12. WebADM Slave nodes configuration (Clustered servers - Enterprise license required)

It is important to fully configure the servers.xml and webadm.conf configuration files on the master before starting the slave setup because configuration files change are not replicated across the cluster when they are changed. That will avoid you edition of same settings multiple times.

To set up the slave nodes, the master WebADM must be started. Each slave nodes must be able to access to the Master on port TCP 5000 (Rsgnd service) and declared in the `/opt/webadm/conf/rsignd.conf` of the master. If these requirements are not met, the slave setup will fail.

```
[root@webadm2 tmp]# /opt/webadm/bin/setup slave
```

```
RCDEVS WEBADM LICENSE AGREEMENT
```

```
RCDevs WebADM Server ("WebADM")
```

```
Copyright (c) 2010-2023 RCDevs Security SA, All rights reserved.
```

```
IMPORTANT: READ CAREFULLY: By using, copying or distributing the Software Product you accept all the following terms and conditions of the present WebADM License Agreement ("Agreement").
```

```
If you do not agree, do not install and use the Software Product.
```

```
WebADM includes additional software products provided by RCDevs SA under freeware and commercial licenses. These additional software are installed under the "/opt/webadm/webapps" and "/opt/webadm/websrvs" directories. This Agreement is subject to all the terms and conditions of any such additional software license.
```

1. DEFINITIONS. "Software Product" means RCDevs Server with which the Agreement is provided which may include third party computer information or software, including apache2, php, libmcrypt, libcurl, libgmp, redis, libxml2, libpng, libqrencode, openldap, openssl, apcu, unixodbc, geoip, expat, hiredis, nghttp2, hiredis, libmaxmind, openscn libcouchbase unmodified software and libraries and related explanatory written materials ("Documentation"). "You" means you or any recipient that obtained a copy of the Software Product pursuant to the terms and conditions of the Agreement.

2. LICENSE. Subject to your compliance with the terms and conditions of the Agreement, including, in particular, the provisions in Sections 3, 5 and 6 below, RCDevs hereby grants You a non-exclusive and royalty-free license to use and distribute the Software Product solely for non-commercial purposes in worldwide. You may:

a. download and install the Software Product on any computer in your possession;

b. use the Software Product and any copy solely for a non-commercial purposes;

c. make any original copies of the Software Product; and

d. distribute any copy of the Software Product only in the form originally furnished by RCDevs with no modifications or additions whatsoever. If You have the slightest doubt that your copy of the Software Product is not original, You must contact RCDevs for an original copy.

3. OBLIGATIONS AND RESTRICTIONS ON LICENSE. The license granted in Section 2 is subject to the following obligations and restrictions:

a. The Software Product and copies are to be used only for non-commercial purposes. Prohibited commercial purposes include, but are not limited to:

```
I agree with RCDevs WebADM terms and conditions (Yes/No): ^[[A^C
[root@webadm2 tmp]# /opt/webadm/bin/setup slave
Checking system architecture...Ok
RCDEVS WEBADM LICENSE AGREEMENT
```

RCDevs WebADM Server ("WebADM")
Copyright (c) 2010-2023 RCDevs Security SA, All rights reserved.

IMPORTANT: READ CAREFULLY: By using, copying or distributing the Software Product you accept all the following terms and conditions of the present WebADM License Agreement ("Agreement").

If you do not agree, do not install and use the Software Product.

WebADM includes additional software products provided by RCDevs SA under freeware and commercial licenses. These additional software are installed under the "/opt/webadm/webapps" and "/opt/webadm/websrvs" directories. This Agreement is subject to all the terms and conditions of any such additional software license.

1. DEFINITIONS. "Software Product" means RCDevs Server with which the Agreement is provided which may include third party computer information or software, including apache2, php, libmcrypt, libcurl, libgmp, redis, libxml2, libpng, libqrencode, openldap, openssl, apcu, unixodbc, geoip, expat, hiredis, nghttp2, hiredis, libmaxmind, openscn libcouchbase unmodified software and libraries and related explanatory written materials ("Documentation"). "You" means you or any recipient that obtained a copy of the Software Product pursuant to the terms and conditions of the Agreement.

2. LICENSE. Subject to your compliance with the terms and conditions of the Agreement, including, in particular, the provisions in Sections 3, 5 and 6 below, RCDevs hereby grants You a non-exclusive and royalty-free license to use and distribute the Software Product solely for non-commercial purposes in worldwide. You may:

a. download and install the Software Product on any computer in your possession;

b. use the Software Product and any copy solely for a non-commercial purposes;

c. make any original copies of the Software Product; and

d. distribute any copy of the Software Product only in the form originally furnished by RCDevs with no modifications or additions whatsoever. If You have the slightest doubt that your copy of the Software Product is not original, You must contact RCDevs for an original copy.

3. OBLIGATIONS AND RESTRICTIONS ON LICENSE. The license granted in Section 2 is subject to the following obligations and restrictions:

a. The Software Product and copies are to be used only for non-commercial purposes. Prohibited commercial purposes include, but are not limited to:

(i) Selling, licensing or renting the Software Product to third parties for a fee (by payment of money or otherwise, whether direct or indirect);

(ii) Using the Software Product to provide services or products to others for which you are compensated in any manner (by payment of money or otherwise, whether direct or indirect), including, without

limitation, providing support or maintenance for the Software Product;

(iii) Using the Software Product to develop a similar application on any platform for commercial distribution.

You shall use your best efforts to promptly notify RCDevs upon learning of any violation of the above commercial restrictions.

b. RCDevs, in its sole and absolute discretion, may have included a portion of the source code or online documentation of the Software. Except for any such portions, YOU SHALL NOT MODIFY, REVERSE ENGINEER, DECOMPILE, DISASSEMBLE, OR OTHERWISE ATTEMPT TO DISCOVER THE SOURCE CODE OF THE SOFTWARE PRODUCT, except to the extent this restriction is prohibited by applicable law. Further, You may not create derivative works of or based on the Software Product.

c. Any copy of the Software Product that you make must conspicuously and appropriately reproduce and contain RCDevs's copyright and other proprietary notices that appear on or in the Software Product (see Software Product for examples of such notices) and disclaimer of warranty; keep intact the Agreement and all notices that refer to the Agreement and any absence of warranty; and give any other recipients of the Software Product a copy of the Agreement.

d. As used in this Agreement, the term "distribute" includes making the Software Product available (either intentionally or unintentionally) to third parties for copying or using. Each time You distribute the Software Product or any original copy of the Software Product, You are responsible for the recipient expressly agree to comply with the terms and conditions of the Agreement. The recipient automatically receives the license to use, copy or distribute the Software Product subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

e. RCDevs shall have no obligation to provide any maintenance, support, upgrades or new releases of the Software Product.

4. INTELLECTUAL PROPERTY OWNERSHIP, RESERVATION OF RIGHTS. Title, copyright, ownership rights, and any other intellectual property rights in and to the Software Product, including its Documentation, and each copy thereof are and shall remain the only and absolute property of RCDevs. Except as expressly stated herein, the Agreement does not grant You any intellectual property rights in the Software Product and all rights not expressly granted are reserved by RCDevs.

5. WARRANTY DISCLAIMER.

THE SOFTWARE PRODUCT IS LICENSED FREE OF CHARGE, AND THERE IS NO WARRANTY OF ANY KIND FOR THE SOFTWARE PRODUCT.

RCDevs PROVIDE THE SOFTWARE PRODUCT "AS IS" WITH ALL FAULTS AND ANY EXPRESS OR IMPLIED WARRANTIES INCLUDING BUT NOT LIMITED TO THE IMPLIED

EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, CUSTOM, ACCURACY OF INFORMATIONAL CONTENT, SYSTEM INTEGRATION OR NON-INFRINGEMENT ARE DISCLAIMED.

THE ENTIRE RISK AS TO THE RESULTS, QUALITY AND PERFORMANCE OF THE SOFTWARE PRODUCT IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU (AND NOT RCDevs) ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

6. LIMITATION OF LIABILITY.

YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT IN NO EVENT WILL RCDevs BE LIABLE FOR ANY DAMAGES, CLAIMS OR COSTS WHATSOEVER INCLUDING ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE, CONSEQUENTIAL OR EXEMPLARY DAMAGES, INCLUDING BUT NOT LIMITED TO, DAMAGES FOR LOSS OF USE, DATA, OR OTHER INTANGIBLE LOSSES, ARISING OUT OF, OR RELATED TO THE AGREEMENT OR TO YOUR USE OR THE INABILITY TO USE THE SOFTWARE PRODUCT OR DOCUMENTATION, EVEN IF RCDevs HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS, DAMAGES OR CLAIMS.

7. TERMINATION. The license granted hereunder is effective until terminated by RCDevs, in its sole discretion, after notification. You may terminate the Agreement at any time by uninstalling and destroying all copies of the Software Product in your possession or control.

This license will terminate automatically if you fail to comply with the terms and conditions of the Agreement above. Upon such termination, you must destroy all copies of the Software Product.

The provisions of Section 5 and 6 shall survive the termination of the Agreement.

8. APPLICABLE LAW AND GENERAL PROVISIONS. The Agreement will be governed by and construed in accordance with the Luxembourg law and submitted to the Luxembourg competent courts.

The URL-link of any open-source files and libraries relating to the Software Product is located in the file docs/licenses.txt.

If you have any questions, notices or information relating to the Agreement, please use the address and contact information included with the Software Product or via the web at <http://www.rcdevs.com/>.

```
I agree with RCDevs WebADM terms and conditions (Yes/No): yes
Enter the master PKI server address: webadm1.support.rcdevs.com
Enter the master PKI server port [5000]: 5000
Enter the master PKI server secret: secret_configured_on_master
Testing PKI server connection... Ok
Retrieving PKI CA certificate...Ok
Reading organization name from CA certificate...
Generating SSL private key... Ok
Creating SSL certificate request... Ok
Signing SSL certificate with PKI server... Ok
```

Certificate and key of the slave node has been generated and signed by Rsignd.

```
Do you want to get configuration from master using SSH (y/n)?
SSH user must have access to /opt/webadm/conf folder of master server!
```

At this step, you have the possibilities to download the configuration by establishing an ssh connection to the master. WebADM configuration files are under root account permissions, then you need to provide root credentials of your master node to download files. If you don't have root credentials, then you must manually copy the following files from the master on the slaves:

- > `/opt/webadm/conf/webadm.conf`
- > `/opt/webadm/conf/servers.xml`
- > `/opt/webadm/conf/objects.xml`
- > `/opt/webadm/conf/license.key`

If you choose yes, then SSH info are asked:

```
Please enter y or n: y
Enter the SSH username: root
Enter the SSH port: 22
Warning: Permanently added 'webadm1.support.rcdevs.com' (ECDSA) to the list of known hosts.
Password:
license.key
100% 992 1.4MB/s 00:00
servers.xml
100% 4739 8.7MB/s 00:00
webadm.conf
100% 11KB 20.3MB/s 00:00
objects.xml
100% 14KB 24.9MB/s 00:00
Setting file permissions... Ok
Adding systemd service... Ok
Adding logrotate scripts... Ok
WebADM has successfully been setup.
```

If you choose no:

```
Please enter y or n: n
Setting file permissions... Ok
Adding systemd service... Ok
Adding logrotate scripts... Ok
WebADM has successfully been setup.
```

In order to finish the configuration of the cluster, please install to the current server the following configuration files from master:

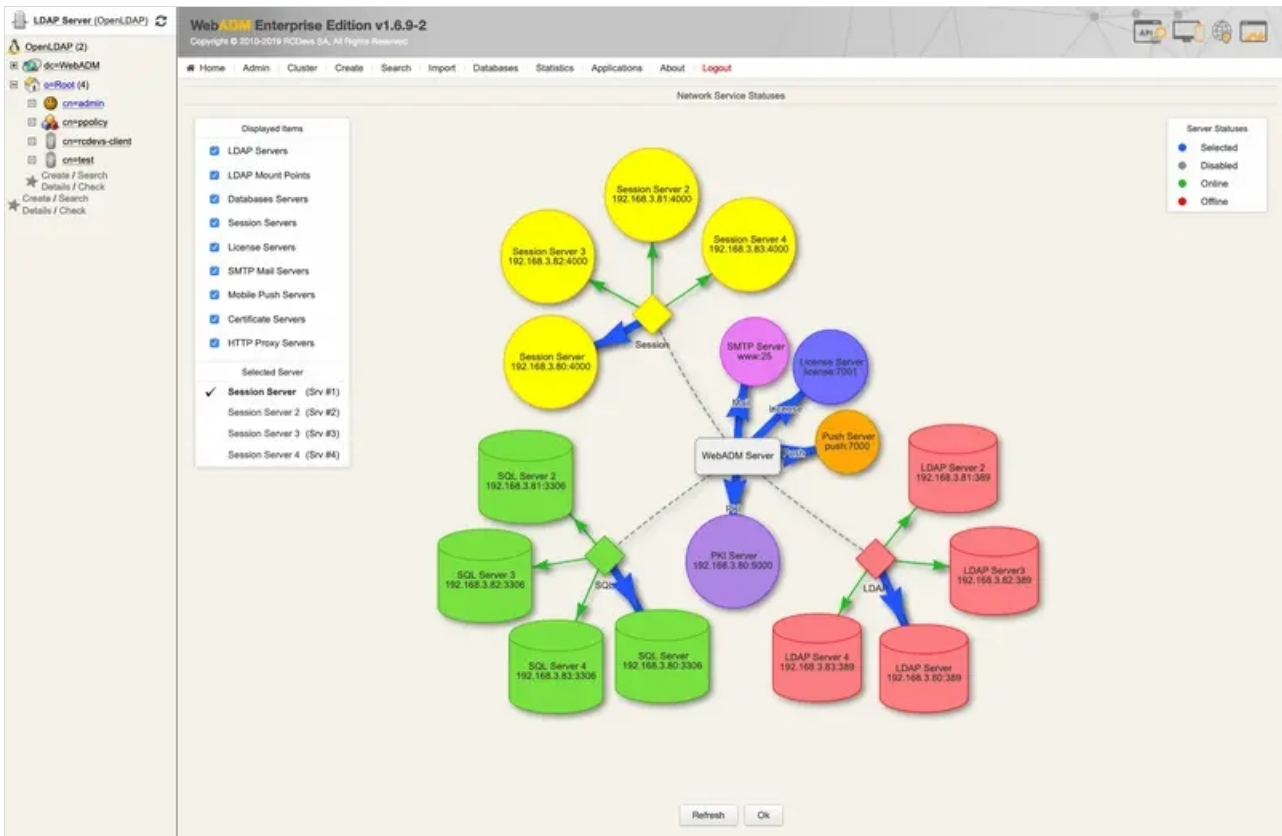
- /opt/webadm/conf/servers.xml
- /opt/webadm/conf/webadm.conf
- /opt/webadm/conf/objects.xml
- /opt/webadm/conf/license.key

13. High Availability of External components

This section demonstrates replication configuration for 4 WebADM, SQL, LDAP (Slapd) nodes.

13.1. CentOS Stream 8 - 4 Nodes

In the following step-by-step example, we will set up a High Availability 4 Nodes Cluster with a MULTI-MASTER MariaDB (TLS) replication and with the RCDevs Directory Server LDAP (TLS) replication.



The HA Cluster will have 4 nodes. The following commands should be run as root. —NODES 1234— means running the commands on every node 1,2,3 and 4.

Warning

Note that you must really do this setup step by step. It will not work if one step is omitted or not following the order.

WebADM requires an accurate system clock, therefore, synchronize the clock. Use `chronyc makestep` for the RCDevs Virtual Appliance and `ntpq -p` if NTP service is used instead.

To simplify the setup can disable the firewall and enable it after having successfully established the replication. Please have a look at [RCDevs Communication Ports](#). It describes the ports and protocols used by RCDevs products between different components. At [RCDevs Hardening Guide](#) is an example of the iptables firewall rules for a high availability cluster with 4 nodes.


```
---NODES 1234---
[root@webadm1 ~]# cat /etc/system-release
CentOS Linux release 7.6.1810 (Core)
[root@webadm1 ~]# yum install chrony
...
Installed:
  chrony.x86_64 0:3.2-2.el7
...
Complete!
[root@webadm1 ~]# systemctl start chronyd
[root@webadm1 ~]# systemctl enable chronyd
[root@webadm1 ~]# chronyc makestep
200 OK
[root@webadm1 ~]# systemctl status chronyd -l
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-02-07 10:28:42 CET; 39s ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
  Main PID: 16580 (chronyd)
    CGroup: /system.slice/chronyd.service
            └─16580 /usr/sbin/chronyd

Feb 07 10:28:42 rcdevs1.webadm1 systemd[1]: Starting NTP client/server...
Feb 07 10:28:42 rcdevs1.webadm1 chronyd[16580]: chronyd version 3.2 starting (+CMDMON +NTP
+REFCLOCK +RTC +PRIVDROP +SCFILTER +SECHASH +SIGND +ASYNCDNS +IPV6 +DEBUG)
Feb 07 10:28:42 rcdevs1.webadm1 chronyd[16580]: Initial frequency -300.000 ppm
Feb 07 10:28:42 rcdevs1.webadm1 systemd[1]: Started NTP client/server.
Feb 07 10:28:47 rcdevs1.webadm1 chronyd[16580]: Selected source 188.42.54.79
Feb 07 10:29:06 rcdevs1.webadm1 chronyd[16580]: System clock was stepped by 0.000002 seconds
[root@webadm1 ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@webadm1 ~]# reboot
```

Be sure that you have a different hostname for each node and put them into `/etc/hosts`. To change the hostname use the command `hostnamectl set-hostname "rcdevs1.webadm1"`.

```
---NODES 1234---  
[root@webadm1 ~]# hostname  
rcdevs1.webadm1  
[root@webadm1 ~]# vi /etc/hosts  
127.0.0.1 localhost  
192.168.4.2 ad1.support.rcdevs.com  
192.168.4.3 ad2.support.rcdevs.com  
192.168.4.4 ad3.support.rcdevs.com  
192.168.4.5 ad4.support.rcdevs.com
```

13.1.1. Directory Server Replication

Use the RCDevs Repository to install the RCDevs Directory Server. The setup script creates the DS system user (slapd), server certificates, filesystem permissions and initializes your LDAP database. During the setup of `/opt/slapd/bin/setup` it will ask to set up an admin password. In this guide, we will use `password` for the LDAP admin password.

```
---NODES 1234---
[root@webadm1 ~]# yum install https://repos.rcdevs.com/redhat/base/rcdevs_release-1.1.1-1.noarch.rpm
...
Installed:
  rcdevs_release.noarch 0:1.1.1-1

Complete!
[root@webadm1 ~]# yum install slapd
...
Installed:
  slapd.x86_64 0:1.0.9-0

Complete!
[root@webadm1 ~]# /opt/slapd/bin/setup
Checking system architecture...Ok
Enter the server fully qualified host name (FQDN): slapd.local
Is this server a standalone LDAP or a replication peer in an LDAP cluster?
Enter 's' for standalone server or 'r' for a replication peer: s
Enter an admin password: Creating self-signed certificate... Ok
Initializing LDAP data... Ok
Setting file permissions... Ok
Starting LDAP Directory... Ok
Setting Admin password... Ok
Do you want LDAP Directory to be automatically started at boot (y/n)? y
Adding systemd service... Ok
Do you want to register LDAP Directory logrotate script (y/n)? y
Adding logrotate script... Ok
Do you want to register LDAP Directory DB backup script (y/n)? y
Adding DB backup script... Ok
LDAP Directory has successfully been setup.
[root@webadm1 ~]#
```

13.1.1.1. Adjust slapd.conf

With RCDevs Directory Server and more generally with OpenLDAP, the replication uses the syncprov overlay. The recommended configuration is a Master-Master Mirror. On the —NODE 1—, edit the `/opt/slapd/conf/slapd.conf` file. Uncomment the replication block, configure it as follows and restart the slapd service.

```
---NODE 1---
[root@webadm1 ~]# vi /opt/slapd/conf/slapd.conf
serverID 1
syncrepl rid=001
provider=ldap://webadm2.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=002
provider=ldap://webadm3.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=003
provider=ldap://webadm4.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
multiprovider on on
[root@webadm1 ~]# /opt/slapd/bin/slapd restart
Stopping RCDevs LDAP Directory... Ok
Checking system architecture... Ok
Checking server configuration... Ok
Starting RCDevs LDAP Directory... Ok
[root@webadm1 ~]#
```

Set up the RCDevs Directory Server for —NODE 234—.

```
---NODE 2---
[root@webadm2 ~]# vi /opt/slapd/conf/slapd.conf
serverID 2
syncrepl rid=001
provider=ldap://webadm1.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=002
provider=ldap://webadm3.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=003
provider=ldap://webadm4.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
multiprovider on
[root@webadm2 ~]# /opt/slapd/bin/slapd restart
Stopping RCDevs LDAP Directory... Ok
Checking system architecture... Ok
Checking server configuration... Ok
Starting RCDevs LDAP Directory... Ok
[root@webadm2 ~]#
```

```
---NODE 3---
[root@webadm3 ~]# vi /opt/slapd/conf/slapd.conf
serverID 3
syncrepl rid=001
provider=ldap://webadm1.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=002
provider=ldap://webadm2.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=003
provider=ldap://webadm4.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
multiprovider on
[root@webadm3 ~]# /opt/slapd/bin/slapd restart
Stopping RCDevs LDAP Directory... Ok
Checking system architecture... Ok
Checking server configuration... Ok
Starting RCDevs LDAP Directory... Ok
[root@webadm3 ~]#
```

```
---NODE 4---
[root@webadm4 ~]# vi /opt/slapd/conf/slapd.conf
serverID 4
syncrepl rid=001
provider=ldap://webadm1.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=002
provider=ldap://webadm2.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
syncrepl rid=003
provider=ldap://webadm3.support.rcdevs.com
bindmethod=simple
binddn="cn=admin,o=root"
credentials="password"
starttls=yes
tls_reqcert=never
searchbase=""
schemachecking=on
type=refreshAndPersist
retry="10 5 60 +"
multiprovider on
[root@webadm4 ~]# /opt/slapd/bin/slapd restart
Checking system architecture... Ok
Checking server configuration... Ok
Starting RCDevs LDAP Directory... Ok
[root@webadm4 ~]#
```

13.1.2. MariaDB Replication

Let's install MariaDB. After having installed MySQL/MariaDB, please run the script called `mysql_secure_installation`. It will ask you to change the root password, remove the ability for anyone to log into MySQL by default, disable logging in remotely

with the administrator account and remove some test databases that are insecure.

Note

This guide also works with the latest CentOS v8.4.2105 and MariaDB v10.3.28.

```
---NODES 1234---
[root@webadm1 ~]# yum install mariadb-server
...
Installed:
  mariadb-server.x86_64 1:5.5.60-1.el7_5
...
Complete!
[root@webadm1 ~]# systemctl start mariadb
[root@webadm1 ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to
/usr/lib/systemd/system/mariadb.service.
[root@webadm1 ~]# mysql_secure_installation
```

**NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!**

In order to log into MariaDB to secure it, we'll need the current password for the root user. If you've just installed MariaDB, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...
```

Setting the root password ensures that nobody can log into the MariaDB root user without the proper authorisation.

```
Set root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

```
Remove anonymous users? [Y/n]
```



```
Remove anonymous users: [Y/n]
```

```
... Success!
```

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
Disallow root login remotely? [Y/n]
```

```
... Success!
```

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n]
```

```
- Dropping test database...
```

```
... Success!
```

```
- Removing privileges on test database...
```

```
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n]
```

```
... Success!
```

```
Cleaning up...
```

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

```
Thanks for using MariaDB!
```

```
[root@webadm1 ~]#
```

13.1.2.1. Adjust server.cnf

Let's set up the MULTI-MASTER MariaDB replication. First edit the MariaDB configuration file `/etc/my.cnf.d/server.cnf`.

Note

For CentOS 8, the MariaDB configuration file is located at `/etc/my.cnf.d/mariadb-server.cnf`.

```
---NODE 1---
[root@webadm1 ~]# vi /etc/my.cnf.d/server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]
bind-address = webadm1.support.rcdevs.com
# Note: Set "server-id" to 1 for Node 1.
server-id = 1
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 4 because there are 4 Nodes.
auto-increment-increment = 4
# Note: Set "auto-increment-offset" to 1 for Node 1.
auto-increment-offset = 1
replicate-do-db = webadm
log_bin = mariadb-bin
log-basename = mariadb
binlog-do-db = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
expire_logs_days = 10

# this is only for embedded server
[embedded]

# This group is only read by MariaDB-5.5 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mysqld-5.5]

# These two groups are only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]

[mariadb-5.5]
[root@webadm1 ~]#
```

```
---NODE 2---
[root@webadm2 ~]# vi /etc/my.cnf.d/server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]
bind-address = webadm2.support.rcdevs.com
# Note: Set "server-id" to 2 for Node 2.
server-id = 2
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 4 because there are 4 Nodes.
auto-increment-increment = 4
# Note: Set "auto-increment-offset" to 2 for Node 2.
auto-increment-offset = 2
replicate-do-db = webadm
log_bin = mariadb-bin
log-basename = mariadb
binlog-do-db = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
expire_logs_days = 10

# this is only for embedded server
[embedded]

# This group is only read by MariaDB-5.5 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mysqld-5.5]

# These two groups are only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]

[mariadb-5.5]
[root@webadm2 ~]#
```

```
---NODE 3---
[root@webadm3 ~]# vi /etc/my.cnf.d/server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]
bind-address = webadm3.support.rcdevs.com
# Note: Set "server-id" to 3 for Node 3.
server-id = 3
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 4 because there are 4 Nodes.
auto-increment-increment = 4
# Note: Set "auto-increment-offset" to 3 for Node 3.
auto-increment-offset = 3
replicate-do-db = webadm
log_bin = mariadb-bin
log-basename = mariadb
binlog-do-db = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
expire_logs_days = 10

# this is only for embedded server
[embedded]

# This group is only read by MariaDB-5.5 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mysqld-5.5]

# These two groups are only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]

[mariadb-5.5]
[root@webadm3 ~]#
```

```
---NODE 4---
[root@webadm4 ~]# vi /etc/my.cnf.d/server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]
bind-address = webadm4.support.rcdevs.com
# Note: Set "server-id" to 4 for Node 4.
server-id = 4
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 4 because there are 4 Nodes.
auto-increment-increment = 4
# Note: Set "auto-increment-offset" to 4 for Node 4.
auto-increment-offset = 4
replicate-do-db = webadm
log_bin = mariadb-bin
log-basename = mariadb
binlog-do-db = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
expire_logs_days = 10

# this is only for embedded server
[embedded]

# This group is only read by MariaDB-5.5 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mysqld-5.5]

# These two groups are only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]

[mariadb-5.5]
[root@webadm4 ~]#
```

Restart the MariaDB service and check its status.

```
---NODES 1234---
[root@webadm1 ~]# systemctl restart mariadb
[root@webadm1 ~]# systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2019-02-07 11:49:52 CET; 27s ago
     Process: 7603 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited,
status=0/SUCCESS)
     Process: 7571 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
    Main PID: 7602 (mysqld_safe)
      CGroup: /system.slice/mariadb.service
              └─7602 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
                  └─7908 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-
dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.log --pid-
file=/var/run/mariadb/mariadb.pid --socket=/var/lib/mysql/mysql.sock

Feb 07 11:49:50 rcdevs1.webadm1 systemd[1]: Starting MariaDB database server...
Feb 07 11:49:50 rcdevs1.webadm1 mariadb-prepare-db-dir[7571]: Database MariaDB is probably
initialized in /var/lib/mysql already, nothing is done.
Feb 07 11:49:50 rcdevs1.webadm1 mariadb-prepare-db-dir[7571]: If this is not the case, make sure the
/var/lib/mysql is empty before running mariadb-prepare-db-dir.
Feb 07 11:49:50 rcdevs1.webadm1 mysqld_safe[7602]: 190207 11:49:50 mysqld_safe Logging to
'/var/log/mariadb/mariadb.log'.
Feb 07 11:49:50 rcdevs1.webadm1 mysqld_safe[7602]: 190207 11:49:50 mysqld_safe Starting mysqld
daemon with databases from /var/lib/mysql
Feb 07 11:49:52 rcdevs1.webadm1 systemd[1]: Started MariaDB database server.
[root@webadm1 ~]# yum install net-tools
...
Installed:
  net-tools.x86_64 0:2.0-0.24.20131004git.el7

Complete!
[root@webadm1 ~]# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.4.20:3306      0.0.0.0:*               LISTEN      7908/mysqld
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      6567/sshd
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      6812/master
tcp        0      0 0.0.0.0:636            0.0.0.0:*               LISTEN      6755/rcdevs-slapd
tcp        0      0 0.0.0.0:389            0.0.0.0:*               LISTEN      6755/rcdevs-slapd
tcp6       0      0 :::22                  :::*                     LISTEN      6567/sshd
tcp6       0      0 :::1:25                 :::*                     LISTEN      6812/master
udp        0      0 127.0.0.1:323         0.0.0.0:*                6250/chronyd
udp6       0      0 :::1:323                :::*                     6250/chronyd
[root@webadm1 ~]#
```

13.1.2.2. Database Replication

WebADM uses a database to store audit logs and localized messages. Application configurations, users and their metadata are directly stored in LDAP rather than in the databases. You must create a webadm database on your SQL server and a webadm user with password webadm, having full permissions on that database.

Let's log in to MariaDB as the root user. Create the webadm user and grant privileges on replication.

```
---NODES 1234---
[root@webadm1 ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE webadm;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT USAGE ON webadm.* to 'webadm'@'localhost' identified by 'webadm';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'localhost';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE USER 'webadm'@'webadm1.support.rcdevs.com' identified by 'webadm';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE USER 'webadm'@'webadm2.support.rcdevs.com' identified by 'webadm';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE USER 'webadm'@'webadm3.support.rcdevs.com' identified by 'webadm';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE USER 'webadm'@'webadm4.support.rcdevs.com' identified by 'webadm';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm1.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm2.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm3.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm4.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm1.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm2.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm3.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm4.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> STOP SLAVE;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
MariaDB [(none)]>
```

```
---NODE 1234---
```

```
MariaDB [(none)]> SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mariadb-bin.000001 | 2215 | webadm      |                    |
+-----+-----+-----+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [(none)]>
```

Warning

The output of `SHOW MASTER STATUS` will reveal the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number.

Let's start with the `—NODE 2—` and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from `—NODE 1—`.

---NODE 2---

```
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm1.support.rcdevs.com', MASTER_USER = 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000001', MASTER_LOG_POS = 2215;
Query OK, 0 rows affected (0.01 sec)
```

MariaDB [(none)]>

Continue with the —NODE 3— and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from —NODE 2—.

---NODE 3---

```
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm2.support.rcdevs.com', MASTER_USER = 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000001', MASTER_LOG_POS = 2215;
Query OK, 0 rows affected (0.01 sec)
```

MariaDB [(none)]>

Continue with the —NODE 4— and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from —NODE 3—.

---NODE 4---

```
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm3.support.rcdevs.com', MASTER_USER = 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000001', MASTER_LOG_POS = 2215;
Query OK, 0 rows affected (0.01 sec)
```

MariaDB [(none)]>

At last the —NODE 1— and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from —NODE 4—.

---NODE 1---

```
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm4.support.rcdevs.com', MASTER_USER  
= 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000001',  
MASTER_LOG_POS = 2215;  
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [(none)]>
```

---NODE 1234---

```
MariaDB [(none)]> START SLAVE;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]>
```

13.1.2.3. Verify Replication Status

---NODE 1---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm4.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000001
Read_Master_Log_Pos: 2215
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 2215
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 4

1 row in set (0.00 sec)

MariaDB [(none)]> exit

Bye

---NODE 2---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm1.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000001
Read_Master_Log_Pos: 2215
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 2215
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
1 row in set (0.00 sec)

MariaDB [(none)]> exit

Bye

---NODE 3---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm2.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000001
Read_Master_Log_Pos: 2215
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 2215
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 2

1 row in set (0.01 sec)

MariaDB [(none)]> exit

Bye

---NODE 4---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm3.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000001
Read_Master_Log_Pos: 2215
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 2215
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 3
1 row in set (0.00 sec)

MariaDB [(none)]> exit

Bye

13.1.3. MariaDB TLS Replication

Let's enable TLS for the MULTI-MASTER MariaDB replication.

Note

This guide also works with the latest CentOS v8.4.2105 and MariaDB v10.3.28.

```
---NODE 1234---  
[root@webadm1 ~]# mkdir /var/lib/mysql/ssl/  
[root@webadm1 ~]# cd /var/lib/mysql/ssl/  
[root@webadm1 ssl]#
```

13.1.3.1. Export Certificates

Instead of using your own certificates, one can issue and export SSL Certificate over WebADM GUI under the Admin tab.

LDAP Server (OpenLDAP)

WebADM Enterprise Edition v1.6.9-3
Copyright © 2010-2019 RCDévs SA, All Rights Reserved

API

Home Admin Cluster Create Search Import Databases Statistics Applications About Logout

WebADM Server Administration

WebADM v1.6.9-3 (64bit) running on server rcddevs1.webadm1 (192.168.3.80) in cluster mode (4 servers). Currently handling 1 connection(s).

Server Version Details: Apache/2.4.38 PHP/7.2.14 OpenSSL/1.0.2q-fips
Internal Server Time: 2019-02-04 14:08:47 Europe/Berlin (NTP check Ok)
Hardware Modules: No HSM Connected
WebADM Features: WebApps (Enabled), WebSrvs (Enabled), Manager (Enabled)

Active LDAP Server: LDAP Server (192.168.3.80) Active SQL Server: SQL Server (192.168.3.80)
Active Session Server: Session Server 2 (192.168.3.81) Active PKI Server: PKI Server (192.168.3.80)

Local Domains (1)

Associate domain names with LDAP user search bases.

Trust Domains (0)

Bridge remote domain names located on distant servers.

Client Policies (0)

Define custom policy settings for consumer applications.

LDAP Mount Points (0)

Connect secondary LDAP servers to the tree view.

LDAP Option Sets (1)

Define LDAP tree constraints for your 'other' administrators.

Administrator Roles (0)

Create admin role templates for your 'other' administrators.

Licensing and Configurations	Runtime Actions
Software License Details	Download WebADM CA Certificate
LDAP Server Details	Download WebADM SSL Certificate
LDAP Server Schema	Issue Server or Client SSL Certificate
Memory Usage Details	Clear Admin Session Cache (1 KB) ⓘ
Hardware Modules Details	Clear WebADM License Cache ⓘ
Remote Manager Interface	Clear WebADM Local Caches (223 KB) ⓘ
Config Object Statuses	Flush WebADM Cluster Caches (1969 KB) ⓘ
Network Service Statuses	Reload WebADM Configurations
WebADM Base Settings	

Click on [Download WebADM CA Certificate](#) to download it.

Now click on [Issue Server or Client SSL Certificate](#), add an **FQDN: mariadbserver** and select **Server**.

LDAP Server (OpenLDAP)

WebADM Enterprise Edition v1.6.9-3
Copyright © 2010-2019 RCDevs SA, All Rights Reserved

API

Home Admin Cluster Create Search Import Databases Statistics Applications About Logout

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component. The certificate is generated with the provided information and signed by WebADM certificate authority.

Main information

Server Hostname (FQDN):

Certificate Type:

Certificate validity (in days):

Private Key Password (optional):

Additional information

Alternative Name(s):

Organization Name:

Organizational Unit:

Country Name:

Locality Name:

State or Province:

Street Address:

Email Address:

Ok Cancel

OpenLDAP (2)

- dc=WebADM
- o=Root (2)
 - cn=admin
 - cn=ppolicy
- Create / Search Details / Check
- Create / Search Details / Check

Download the Key and Cert File.



Create Third-party SSL Server Certificate

Creating private key... Success

Certificate details:

- commonName: mariadbserver
- description: SERVER

Creating a certificate request based on the above details... Success

Calling WebADM CA for certificate request signing... Success

Private Key (PEM format):

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQDITA9RONNoemle
e9RvP+tzurTQcc4hFtKv0Hhp7XcKEkbylcoJ5kol+hqd8pDOQgA0778s/PsDDB7
NRtOKPaZrV2YBokWmMy4B0+nckpVL6umZqYacpo40uAkGYqsJg69hyqfH728xsRF
zCDbrLyPXJhEsrSMkjUaeMjJICi4zGrh90JqV9J1IzJpFs5N317qVrUFgKgDZwzU
y4wuSCZAEM2iiCbzQ8D89dljPX1uauUgIK8pLaenCL/MJttKMRPTLe63Z6NnIq0H
j4kgbl9xxyX/Lj3bLcm5eV3qNjfiNm/WQ9poe2hWvXZISXolSfFicjwwXaOccGC
PI1UeBg9AgMBAECCggEAdikYqCvd29DLNGgmZiYc1EUCwhDr/BVo3dGyhfkWQG6r
om9dtlyU6B14DNgaREnyxLo142IttcUDHMwZjaRwxfwWSMYs053Ut4Poo7myzEpR
Ettfhh6xiHaut10zB9TOMVT5+eF5x1HjG+HuSGnA2zoewfhCtuLR3QoAOVZxmdtC
sSYxtHh8VRDSfIq0hABv/pKLXWVfBauTDSYyb5LPwMirfFkPpPTpeX7U4D5PMJE2
8i3Hc5/Rt0mux5KP5qy4AlGyz15zvZUghw3RqH96SCEhMLXJBWTo7bIQMHZKbTR2
3iAZGrSSfyRTmQCEXhS1nZPgunklg6eR5NVJpQTJaQKBgQDpKNs1KqNm/7YASFsb
roCotw8PXxYFEqHbFxlvd2dJgwhkZiPHE0mNs+6/5bnx8Z5FLbfY2xDk9Y6ZmqH
qRwTtetEQvhs9plG6gVqWDy9H0jQT6br3Xz81ARiBxG8E+NkNU617onf0sTmQJ
VXGH985ReOFEPypHWLlowoLUzWKBgQDb6xU8STZDB+ZEHbs2MYbCLbN1ZKOKMIE5
//CRHXARFES1t1Ix981c8/5+4mQzhyXV7SJMt6FH8+vh8btFTi1I/hP6FZs2SUor
UiAGqhkrTLGGbgnbq54lgBADjff+lyRcgp6zlmihiiV6aqIG+PSqq/DHKYSU6V4I
cTDG8H/dmWKBgCR/i8aGpSvTbdcffHuZ1nQsQ93ZIOrrh8C3HpmI3V8jPJDWvc1l
wbHwtJXuOp4vzKbZkT5tXt45PzjHXXf9BfMTqHd4EFIuqxHJwmlCQMfNxbgOL/AO
-----END PRIVATE KEY-----
```

Certificate (PEM format):

```
-----BEGIN CERTIFICATE-----
MIIDBzCCAe+gAwIBAgIBBzANBgkqhkiG9w0BAQsFADALMRIWEAYDVQQDDAlXZWJB
RE0gQ0ExDzANBgNVBAoMB1JDRGV2czAeFw0xOTAyMDQxMzI3NDJhFw0yOTAyMDEx
MzI3NDJhMCKxYjAUBGNVBAAMDDWlhcmllhZGJzZXJ2ZXIxDzANBgNVBAoMB1NFU1Z
UjCCAS1wDQYJK4ZIhvcNAQEBBQADggEPADCCAQoCggEBAMhMD1E402h6aV571G8/
6306tNBBxzIeW0q/QeGntdwoSRvKVygnmSjX6Gp3ykM5CADTvvvyz8+wMMHs1G04o
9pmtXZgE6RaYzLgHT6dySlUvq6ZmpHPymjjs4gQZiqyODr2HKp8fvbzGxEXMINus
vI9cmESytIySNRp4yMkgIjjMauH3QmpX0nUjMmkWzk3fxupWtQWAqANnDNTLjC5I
JkAQzaIqJvNDwPz13WM9fW5q5SAgryktp6cIv8wm20oxE9Mt7rdno2cidQePiSBu
X3GzJf8uPdssKb15Xeo2N8g2b9ZD2mh6zaFa9dkhJejVLAUhyPDBdo5xwYI8jVR4
GD0CAwEAAM+MDwGAYDVR0RBBEwD4INbWFyaWFkYnNlcnZlcjALBgNVHQ8EBAMC
A6gwEwYDVR0lBAwwCgYIKwYBBQUHAWEdQYJKoZIhvcNAQELBQADggEBACgeUM10
tkLDRzYMK1GoQTTmMwciqx6E6vHYtnt79tqD2yTzdxCM4Icv5wAYrflYrSOugUwKj
RLhC+mzBbxP3d2wHukfP1DEjjYnjCe6pHGfHhghy/M5jqgHQZzUrWgSSupzqNW8W
KgNpJhws7GzqSowplh36uPBTopUYQcax/p9b1S5Qf7Xm13gu07BwEMfb8401gxWt
raMv30QuB+0Hiyin9Gg2xnqWkpOm5pU5K9mE0UKW+hRXDt95gxLd0sdwFMie+sSR
Onc5VvvcQSIR3h6JtwExPpC4rTz8kGsfNurUmlzRvjs+4+xHV6EtK5SIkahn+gn6
IZcIQnPY4E19Acc=
-----END CERTIFICATE-----
```

Click on Issue Server or Client SSL Certificate, add an FQDN: mariadbclient and select Client.

LDAP Server (OpenLDAP)

OpenLDAP (2)

- dc=WebADM
- o=Root (2)
 - cn=admin
 - cn=ppolicy
- Create / Search Details / Check
- Create / Search Details / Check

WebADM Enterprise Edition v1.6.9-3
Copyright © 2010-2019 RCDevs SA, All Rights Reserved

API

Home Admin Cluster Create Search Import Databases Statistics Applications About Logout

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component. The certificate is generated with the provided information and signed by WebADM certificate authority.

Main information

Client Name or Description:

Certificate Type:

Restricted Application:

Certificate validity (in days):

Private Key Password (optional):

Additional information

Organization Name:

Organizational Unit:

Country Name:

Locality Name:

State or Province:

Street Address:

Email Address:

Ok Cancel

Download Cert & Key File.

Create Third-party SSL Server Certificate

Creating private key... **Success**

Certificate details:

- commonName: **mariadbclient**- description: **CLIENT**Creating a certificate request based on the above details... **Success**Calling WebADM CA for certificate request signing... **Success**

Private Key (PEM format):

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCsfzyZG103tSxG
H8rq1ldc3GFcxSdqkeqZl0ey2mqNkM9UgJzLC2PktVRKMVeG5cSHZS7BOA3w1zIC
cFiPrPyUVU72xRqXkL8tren5iLOKRNOra4dpgmFz1flki/zhcf2EkXYh+uDCtygd
K5rI/81PqQ2U8jfsztUiwtnFDw2Xvwwk4Uz6hpm1tRDBBYI+jy49kkAHfBAtFj8L
oHJYsqEaTW6QKQZa82AQoTqVGXNfrMe35KBJU8uKBCi9x9CE/hqyHY/VqSvEv4R
BvhJozYvJlN01vzPphMQazt8V0d7Qu7rjk47ChHhLS3cIMNOjn/7CM0cljAkkIGZ
01cylpTXAgMBAECGgEBAI4LTIxLs0xfXXD1VzRggoTlUyuB43SF59nC8GDaaUPf
/kr52tmbLIw2IY3i6c+Ev3w6/vBPR3NaHe1SNCTd6M4BHcApeQdcQ4HLF8PLJQV0
PdUyV7yzh6oV0sNQX1IKexhCbEP5AkJ1cJryORQ+BbkaXbvMZe8UNWcj8Y3LIoDU
h4wPHYMDLdN5JSKB22KpOfIMxdjEGym014hmftU60lhUVUZde46T4swPDKOFih3a
fQe818RL6S+A2h1RyOtArG8AWXkOAr1FZ5hr4RpCbVb1lKQWCv2x2A1HHIqDEJ2S
19MBBjEpkDOI9o74d+Xq8rRq7wJZsqStd5/Gkcfh3FkCgYEA0YTXkx8Y9XsL7cen
NQHGxhok0GMxsAakv7EhJ+8zIz2lZw549f6BtgW4+VG8IIqUaV218VYorKSDLMdc
KybeS1+SfSgfConPvNvJzaIoZurs/V/kZZ+6nL+tIEGvjrEfo9tLeTO/wopiqLeV
iyUpvB5sWSFRK6Fy4JUmvkVzGsCgYEA0SSQym0Q4ad3cbMTSUSDMIXqazR0jJvx
yLbjk0kDZmqN6H/dBHE3Dhd3JTexbGNj1tel23OqrCLaLA8EyreZ9lZeg8QuVlj
qmM9bFMS2ZUdzfI2I0q3MXrYsMnTJ2wnuCPQWnpibCf65mgaROpezxG5zmfziKYy
/PfDP6eYdEUCgYAbFC2Fk8ZqrTYxb5qVJL4hEW9JPn9/2MpGo1xGN9oyGdTdq7B9
s/9EreU0LHRZJAY2mqr9nTgoNBBoTsUWT6zQs9ggrwWUH2p05yPo+HQABvu6PRMY
-----
```

Certificate (PEM format):

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAdWgAwIBAgIBBjANBgkqhkiG9w0BAQsFADAlMRIwEAYDVQQDDAlXZWJB
REoGQ0ExDzANBgNVBAoMB1JDRGV2czAeFw0xOTAyMDQxMzE3MDlaFw0yNDYMDMx
MzE3MDlaMCKxYjAiBgNVBAMMDW1hcmllhZGJjbGllbnQxZANBgNVBAoMBkNMSUVO
VDCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBBAKx/PJkbXTelLEYfyurW
V0LcYVzGwOqR6BmXR7Laao2Qz1SCNksLY+S1VEoxV4blxIdlLsE4DfCXMGJwWI+s
/JRVTVbFGpeQvy2t6fmIs4pE06sDh2mCYXOV+WSL/OFx/YSRdiH64MK3Kp0rkgj/
yU+pDZTyN+z01SLC2cUPDZe/CSThTPqGkyW1EMMFgJ6PLj2Sgfd8EC0WPwugcliy
oRpNavpApBkdZyBChOpUZcl+sx7fkoElTy4oEKL3H0IT+GrIdj9WpJV5XhEG+Emj
Ni8mU3TW/M+mExBrO3xXR3tC7uuOTjsKEeEtLdwgW060F/sIzRyWMCSQgZnTVzKW
lNcCAwEAAMkMCiCwYDVR0PBAQDAgOIMBMGA1UdJQQMMAoGCCsGAQUFBwMCA0G
CSqGSIB3DQEBChUAA4IBAQCnMXNj5hejGs0V9zRs2K7JtKI/2Mn7CmmdGANriHua
JqrfWXXI3r7NLND6igk0ZKa7SFF9/tsQsSfqGhC746z4aUDD/6AobNXUNzt11B+O
DItg+X11EGZ0nMV3K0aBHyR15AF80ahqIUfVeF3WfaNFRXs/ANmQNZE8znrYyT1
syi5su609wUVbhjF7B6XDmaVh/jzv/NG4G842PS9SV5oRNDCh2x6rsa8SZQ03D
V4EzR/lpfr9Bu3b39rWle8NrAjGD3JlJgsY83oTiow2+OastPEEN2AZhgtURHVX
00YdtT6RMVykWdTUC8Fpqu6VIXULNhnTPOLalgb+FFw1
-----END CERTIFICATE-----
```


Finally, copy the following certificates `ca.crt` and `mariadbclient.crt` in to the WebADM configuration folder `/opt/webadm/conf/`.

13.1.3.2. Copy Certificates to all the Nodes

Copy the following certificates `ca.crt`, `mariadbserver.crt` and `mariadbserver.key` to all the nodes —NODE 1234

```
administor:Downloads$ ssh root@webadm1.support.rcdevs.com mkdir /tmp/ssl/
root@webadm1.support.rcdevs.com's password:
administor:Downloads$ ssh root@webadm2.support.rcdevs.com mkdir /tmp/ssl/
root@webadm2.support.rcdevs.com's password:
administor:Downloads$ ssh root@webadm3.support.rcdevs.com mkdir /tmp/ssl/
root@webadm3.support.rcdevs.com's password:
administor:Downloads$ ssh root@webadm4.support.rcdevs.com mkdir /tmp/ssl/
root@webadm4.support.rcdevs.com's password:
administor:Downloads$ scp *.pem root@webadm1.support.rcdevs.com:/tmp/ssl/
root@webadm1.support.rcdevs.com's password:
ca.crt                100% 1142   1.7MB/s  00:00
mariadbserver.crt    100% 1092   1.5MB/s  00:00
mariadbserver.key    100% 1092   1.4MB/s  00:00
administor:Downloads$ scp *.pem root@webadm2.support.rcdevs.com:/tmp/ssl/
root@webadm2.support.rcdevs.com's password:
ca.crt                100% 1142   1.6MB/s  00:00
mariadbserver.crt    100% 1092   1.6MB/s  00:00
mariadbserver.key    100% 1092   1.6MB/s  00:00
administor:Downloads$ scp *.pem root@webadm3.support.rcdevs.com:/tmp/ssl/
root@webadm3.support.rcdevs.com's password:
ca.crt                100% 1142   1.5MB/s  00:00
mariadbserver.crt    100% 1092   1.5MB/s  00:00
mariadbserver.key    100% 1092   1.4MB/s  00:00
administor:Downloads$ scp *.pem root@webadm4.support.rcdevs.com:/tmp/ssl/
root@webadm4.support.rcdevs.com's password:
ca.crt                100% 1142   1.6MB/s  00:00
mariadbserver.crt    100% 1092   1.4MB/s  00:00
mariadbserver.key    100% 1092   1.6MB/s  00:00
administor:Downloads$
```

Warning

Set the owner to `mysql` and the rights for the MariaDB certificate files.

```
---NODE 1234---
[root@webadm1 ssl]# mv /tmp/ssl/* /var/lib/mysql/ssl/
[root@webadm1 ssl]# chown mysql:mysql /var/lib/mysql/ssl/
[root@webadm1 ssl]# chown mysql:mysql /var/lib/mysql/ssl/*
[root@webadm1 ssl]# chmod 640 /var/lib/mysql/ssl/*
[root@webadm1 ssl]# rm -r /tmp/ssl/
[root@webadm1 ssl]#
```

13.1.3.3. Adjust server.cnf

Edit the MariaDB configuration file `/etc/my.cnf.d/server.cnf` on all the nodes —NODE 1234— to add the path of the certificates, `ssl-ca`, `ssl-cert` and `ssl-key`. Afterward, restart the MariaDB service.

Note

For CentOS 8, the MariaDB configuration file is located at `/etc/my.cnf.d/mariadb-server.cnf`.

```
---NODE 1234---
[root@webadm1 ssl]# vi /etc/my.cnf.d/server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]
bind-address = webadm1.support.rcdevs.com
server-id = 1
replicate-same-server-id = 0
auto-increment-increment = 4
auto-increment-offset = 1
replicate-do-db = webadm
log_bin = mariadb-bin
log-basename = mariadb
binlog-do-db = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
expire_logs_days = 10
ssl-ca=/var/lib/mysql/ssl/ca.crt
ssl-cert=/var/lib/mysql/ssl/mariadbserver.crt
ssl-key=/var/lib/mysql/ssl/mariadbserver.key
...

[root@webadm1 ssl]# systemctl restart mariadb
[root@webadm1 ssl]# systemctl status mariadb -l
● mariadb.service - MariaDB database server
```

```

Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
Active: active (running) since Thu 2019-02-07 13:39:53 CET; 3s ago
Process: 24381 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited,
status=0/SUCCESS)
Process: 24349 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
Main PID: 24380 (mysqld_safe)
CGroup: /system.slice/mariadb.service
├─24380 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
└─24722 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-
dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.log --pid-
file=/var/run/mariadb/mariadb.pid --socket=/var/lib/mysql/mysql.sock

```

```

Feb 07 13:39:51 rcdevs1.webadm1 systemd[1]: Starting MariaDB database server...
Feb 07 13:39:51 rcdevs1.webadm1 mariadb-prepare-db-dir[24349]: Database MariaDB is probably
initialized in /var/lib/mysql already, nothing is done.
Feb 07 13:39:51 rcdevs1.webadm1 mariadb-prepare-db-dir[24349]: If this is not the case, make sure the
/var/lib/mysql is empty before running mariadb-prepare-db-dir.
Feb 07 13:39:51 rcdevs1.webadm1 mysqld_safe[24380]: 190207 13:39:51 mysqld_safe Logging to
'/var/log/mariadb/mariadb.log'.
Feb 07 13:39:51 rcdevs1.webadm1 mysqld_safe[24380]: 190207 13:39:51 mysqld_safe Starting mysqld
daemon with databases from /var/lib/mysql
Feb 07 13:39:53 rcdevs1.webadm1 systemd[1]: Started MariaDB database server.

```

```
[root@webadm1 ssl]# netstat -tulpn
```

```
Active Internet connections (only servers)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:5000	0.0.0.0:*	LISTEN	8171/webadm-rsighnd
tcp	0	0	192.168.4.20:3306	0.0.0.0:*	LISTEN	24722/mysqld
tcp	0	0	0.0.0.0:8080	0.0.0.0:*	LISTEN	8217/webadm-httpd
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN	8217/webadm-httpd
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	6567/sshd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	6812/master
tcp	0	0	0.0.0.0:8443	0.0.0.0:*	LISTEN	8217/webadm-httpd
tcp	0	0	0.0.0.0:443	0.0.0.0:*	LISTEN	8217/webadm-httpd
tcp	0	0	0.0.0.0:636	0.0.0.0:*	LISTEN	6755/rcdevs-slapd
tcp	0	0	0.0.0.0:4000	0.0.0.0:*	LISTEN	8164/webadm-session
tcp	0	0	0.0.0.0:389	0.0.0.0:*	LISTEN	6755/rcdevs-slapd
tcp6	0	0	:::22	:::*	LISTEN	6567/sshd
tcp6	0	0	:::1:25	:::*	LISTEN	6812/master
tcp6	0	0	:::4000	:::*	LISTEN	8164/webadm-session
udp	0	0	127.0.0.1:323	0.0.0.0:*		6250/chronyd
udp6	0	0	:::1:323	:::*		6250/chronyd

```
[root@webadm1 ssl]#
```

13.1.3.4. Enable SSL/TLS

Log in to MariaDB as the root user and enable the SSL/TLS.

---NODE 1234---

```
[root@webadm1 ssl]# mysql -u root -p
```

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 102

Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'localhost' REQUIRE X509;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm1.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm2.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm3.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm4.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'localhost' REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm1.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm2.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm3.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm4.support.rcdevs.com'  
REQUIRE SSL;  
Query OK, 0 rows affected (0.00 sec)
```



```
MariaDB [(none)]> STOP SLAVE;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]>
```

```
---NODE 1234---  
MariaDB [(none)]> SHOW MASTER STATUS;  
+-----+-----+-----+-----+  
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |  
+-----+-----+-----+-----+  
| mariadb-bin.000002 | 1739 | webadm      |                    |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

```
MariaDB [(none)]>
```

Warning

The output of `SHOW MASTER STATUS` will reveal the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number.

Let's start with the `—NODE 2—` and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from `—NODE 1—`.

```
---NODE 2---  
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm1.support.rcdevs.com', MASTER_USER  
= 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000002',  
MASTER_LOG_POS = 1739, MASTER_SSL=1;  
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]>
```

Continue with the `—NODE 3—` and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from `—NODE 2—`.

```
---NODE 3---  
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm2.support.rcdevs.com', MASTER_USER  
= 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000002',  
MASTER_LOG_POS = 1739, MASTER_SSL=1;  
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [(none)]>
```

Continue with the `---NODE 4---` and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from `---NODE 3---`.

```
---NODE 4---
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm3.support.rcdevs.com', MASTER_USER
= 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000002',
MASTER_LOG_POS = 1739, MASTER_SSL=1;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
```

At last the `---NODE 1---` and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from `---NODE 4---`.

```
---NODE 1---
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = 'webadm4.support.rcdevs.com', MASTER_USER
= 'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mariadb-bin.000002',
MASTER_LOG_POS = 1739, MASTER_SSL=1;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>

---NODE 1234---
MariaDB [(none)]> START SLAVE;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
```

13.1.3.5. Verify TLS Status

Verify MariaDB TLS as follows:

---NODE 1---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm4.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000002
Read_Master_Log_Pos: 1739
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000002
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1739
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: Yes
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 4

1 row in set (0.00 sec)

MariaDB [(none)]> exit

Bye

---NODE 2---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm1.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000002
Read_Master_Log_Pos: 1739
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000002
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1739
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: Yes
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
1 row in set (0.00 sec)

MariaDB [(none)]> exit

Bye

---NODE 3---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm2.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000002
Read_Master_Log_Pos: 1739
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000002
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1739
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: Yes
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 2
1 row in set (0.00 sec)

MariaDB [(none)]> exit

Bye

---NODE 4---

MariaDB [(none)]> SHOW SLAVE STATUS \G

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: webadm3.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000002
Read_Master_Log_Pos: 1739
Relay_Log_File: slave-relay.000002
Relay_Log_Pos: 531
Relay_Master_Log_File: mariadb-bin.000002
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1739
Relay_Log_Space: 821
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: Yes
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 3
1 row in set (0.00 sec)

MariaDB [(none)]> exit

Bye

```
---NODE 1234---
MariaDB [(none)]> SHOW VARIABLES LIKE '%ssl%';
+-----+-----+
| Variable_name | Value                               |
+-----+-----+
| have_openssl  | YES                                 |
| have_ssl      | YES                                 |
| ssl_ca        | /var/lib/mysql/ssl/ca.crt         |
| ssl_capath    |                                     |
| ssl_cert      | /var/lib/mysql/ssl/mariadbserver.crt |
| ssl_cipher    |                                     |
| ssl_key       | /var/lib/mysql/ssl/mariadbserver.key |
+-----+-----+
7 rows in set (0.00 sec)
```

Do a local authentication with the previously built certifications to verify that they are good.

```
[root@lolocentos7a ~]# mysql -u root -p --ssl-ca=/var/lib/mysql/ssl/ca.crt --ssl-
cert=/var/lib/mysql/ssl/mariadbserver.crt --ssl-key=/var/lib/mysql/ssl/mariadbserver.key
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 164
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> status;
-----
mysql Ver 15.1 Distrib 5.5.68-MariaDB, for Linux (x86_64) using readline 5.1

Connection id: 164
Current database:
Current user: root@localhost
SSL:  Cipher in use is DHE-RSA-AES256-GCM-SHA384
Current pager: stdout
Using outfile: ''
Using delimiter: ;
Server:  MariaDB
Server version:  5.5.68-MariaDB MariaDB Server
Protocol version: 10
Connection:  Localhost via UNIX socket
Server characterset: latin1
Db  characterset: latin1
Client characterset: latin1
Conn. characterset: latin1
UNIX socket:  /var/lib/mysql/mysql.sock
Uptime:  19 min 0 sec

Threads: 2  Questions: 223  Slow queries: 0  Opens: 8  Flush tables: 2  Open tables: 34  Queries per
second avg: 0.195
-----
```

Check that `SSL: Cipher in use...` is present.

13.1.3.6. Adjust servers.xml

Finally, adjust the parameter encryption from `NONE` to `TLS` and add the certificates in the configuration file `/opt/webadm/conf/servers.xml` of all nodes—NODE 1234—. Afterward, restart WebADM to enable TLS for MULTI-MASTER MariaDB replication.

Note

Please use the **MySQL8** instead of the **MariaDB** driver for certificate based authentication.

```
---NODE 1234---
[root@webadm1 ssl]# vi /opt/webadm/conf/servers.xml
<SqlServer name="SQL Server"
type="MySQL8"
host="webadm1.support.rcdevs.com"
user="webadm"
password="webadm"
database="webadm"
encryption="TLS"
ca_file="/opt/webadm/conf/ca.crt"
cert_file="/opt/webadm/conf/mariadbclient.crt"
key_file="/opt/webadm/conf/mariadbclient.key"/>
<SqlServer name="SQL Server 2"
type="MySQL8"
host="webadm2.support.rcdevs.com"
user="webadm"
password="webadm"
database="webadm"
encryption="TLS"
ca_file="/opt/webadm/conf/ca.crt"
cert_file="/opt/webadm/conf/mariadbclient.crt"
key_file="/opt/webadm/conf/mariadbclient.key"/>
<SqlServer name="SQL Server 3"
type="MySQL8"
host="webadm3.support.rcdevs.com"
user="webadm"
password="webadm"
database="webadm"
encryption="TLS"
ca_file="/opt/webadm/conf/ca.crt"
cert_file="/opt/webadm/conf/mariadbclient.crt"
key_file="/opt/webadm/conf/mariadbclient.key"/>
<SqlServer name="SQL Server 4"
type="MySQL8"
host="webadm4.support.rcdevs.com"
user="webadm"
password="webadm"
database="webadm"
encryption="TLS"
ca_file="/opt/webadm/conf/ca.crt"
cert_file="/opt/webadm/conf/mariadbclient.crt"
key_file="/opt/webadm/conf/mariadbclient.key"/>
```

```
[root@webadm1 ssl]# /opt/webadm/bin/webadm restart
Stopping WebADM HTTP server... Ok
Stopping WebADM Watchd server..... Ok
Stopping WebADM PKI server... Ok
Stopping WebADM Session server... Ok
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Ok

Found Trial Enterprise license (LOIC)
Licensed by RCDevs SA to LOIC
Licensed product(s): OpenOTP

Starting WebADM Session server... Ok
Starting WebADM PKI server... Ok
Starting WebADM Watchd server... Ok
Starting WebADM HTTP server... Ok

Checking server connections. Please wait...
Connected LDAP server: LDAP Server (webadm1.support.rcdevs.com)
Connected SQL server: SQL Server (webadm1.support.rcdevs.com)
Connected PKI server: PKI Server (webadm2.support.rcdevs.com)
Connected Session server: Session Server 2 (webadm2.support.rcdevs.com)

Checking LDAP proxy user access... Ok
Checking SQL database access... Ok
Checking PKI service access... Ok

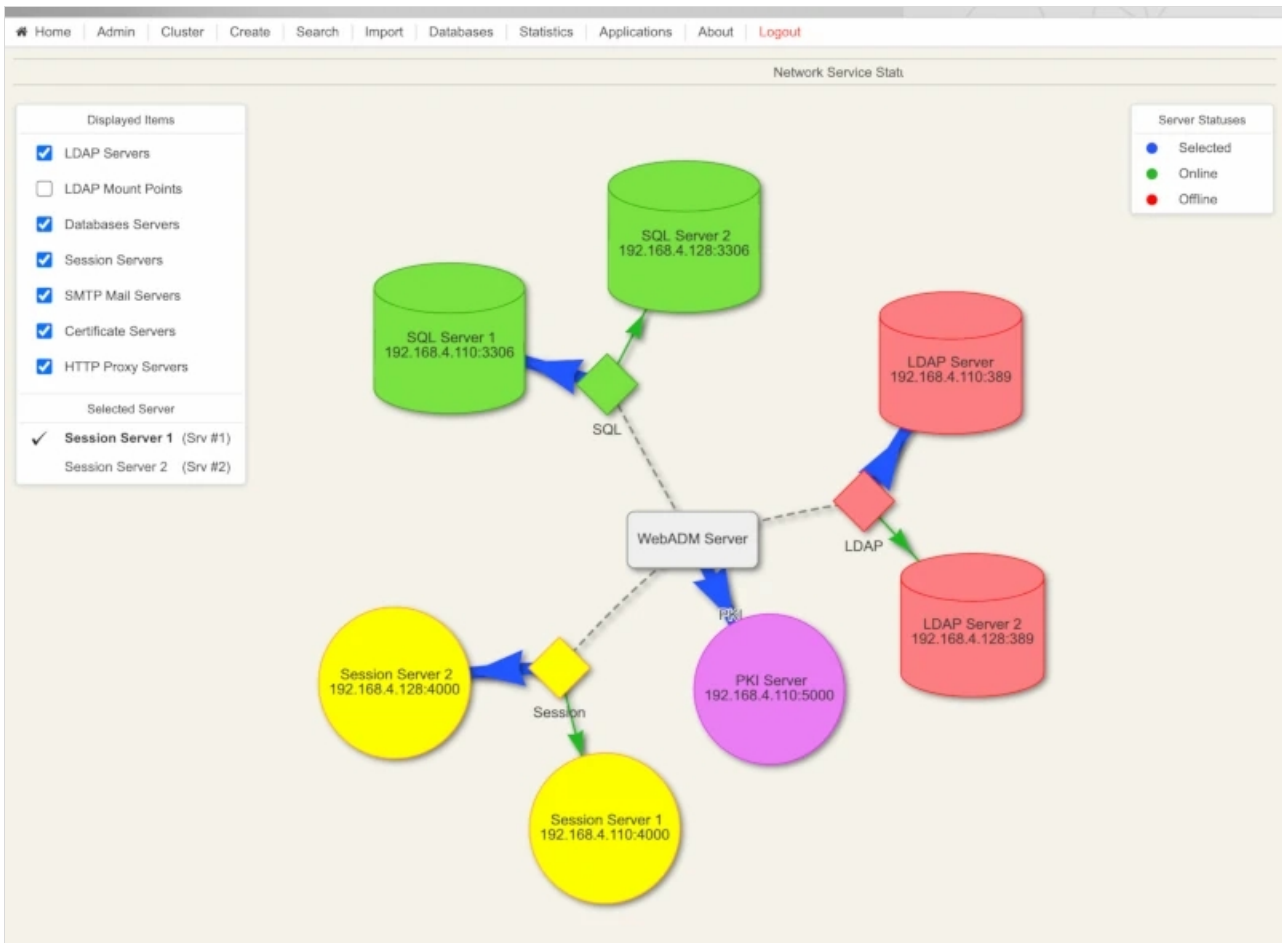
Cluster mode enabled with 4 nodes (I'm slave)
Session replication status: Active (0.0014 sec)
[root@webadm1 ssl]#
```

13.2. Ubuntu 22.04 with MySQL8 TLS Replication - 2 Nodes

This documentation has been tested on [Ubuntu 22.04](#) with [MySQL8](#), and we are using **2 Nodes**:

> **Node 1: webadm1.support.rcdevs.com**

> **Node 2: webadm2.support.rcdevs.com**



Please note that this documentation is not a guide to install WebADM, SlapD or even MySQL8. For this please refer to the following links:

- > [RCDevs Directory Server Installation \(SlapD\)](#)
- > [WebADM](#)
- > [MySQL8 Quick Guide \(apt\)](#)

To have a functional replication using **Transport Layer Security (TLS)**, you need to configure your **WebADM**, **Slapd** and **MySQL8** servers as explained below:

13.2.1. Adjust slapd.conf:

---NODE 1---

```
root@webadm1: nano /opt/slapd/conf/slapd.conf
```

```
serverID 1
```

```
syncrepl rid=001
```

```
    provider=ldap://webadm2.support.rcdevs.com
```

```
    bindmethod=simple
```

```
    binddn="cn=admin,o=root"
```

```
    credentials="password"
```

```
    starttls=yes
```

```
    tls_reqcert=never
```

```
    searchbase=""
```

```
    schemachecking=on
```

```
    type=refreshAndPersist
```

```
    retry="10 5 60 +"
```

```
multiprovider on
```

---NODE 2---

```
root@webadm2: nano /opt/slapd/conf/slapd.conf
```

```
serverID 2
```

```
syncrepl rid=001
```

```
    provider=ldap://webadm1.support.rcdevs.com
```

```
    bindmethod=simple
```

```
    binddn="cn=admin,o=root"
```

```
    credentials="password"
```

```
    starttls=yes
```

```
    tls_reqcert=never
```

```
    searchbase=""
```

```
    schemachecking=on
```

```
    type=refreshAndPersist
```

```
    retry="10 5 60 +"
```

```
multiprovider on
```

---NODES 12---

```
/opt/slapd/bin/slapd restart
```

13.2.2. WebADM configuration

Rsignd configuration

Rsignd is the PKI service running with WebADM. The configuration of Rsignd is located in

```
/opt/webadm/conf/rsignd.conf:
```

```

---NODE 12---
root@webadm1: nano /opt/webadm/conf/rsignd.conf

# Declare here the Rsign clients with IP addresses or hostnames.
# In cluster mode, the client WebADM server(s) must be defined here!
client {
    hostname webadm1.support.rcdevs.com
    secret secret
}

client {
    hostname webadm2.support.rcdevs.com
    secret secret
}

```

Servers.xml

```

---NODE 12---
nano /opt/webadm/conf/servers.xml

<Servers>

<!--
*****
*** WebADM Remote Server Connections ***
*****

```

You can configure multiple instances for each of the following servers. At login, WebADM will try to connect the configured servers in the same order they appear in this file and uses the first one it successfully establishes the connection to. If the server connection goes down, it will automatically failover to the next configured server.

Any special characters must be encoded in XML compliant format. At least one LDAP server and one SQL server is required to run WebADM. Supported servers: OpenLDAP, Active Directory, Novell eDirectory, 389.

Allowed LDAP parameters are:

- name: server friendly name
- host: server hostname or IP address
- port: LDAP port number
default and TLS: 389
default SSL: 636
- encryption: connection type
allowed type are NONE, SSL and TLS
default: 'NONE'
- ca_file: Trusted CA for SSL and TLS

- cert_file: client certificate file
- key_file: client certificate key

-->

```
<LdapServer name="LDAP Server 1"  
  host="webadm1.support.rcdevs.com"  
  port="389"  
  encryption="TLS"  
  ca_file="" />
```

```
<LdapServer name="LDAP Server 2"  
  host="webadm2.support.rcdevs.com"  
  port="389"  
  encryption="TLS"  
  ca_file="" />
```

<!--

SQL servers are used for logs; message localizations and inventories.

Supported servers: MySQL5, MySQL8, PostgreSQL, MSSQL, Sybase, Oracle, SQLite.

Allowed SQL parameters are:

- type: MySQL5, MySQL8, MariaDB, PostgreSQL, MSSQL, SQLite.
- name: server friendly name
- host: server hostname or IP address
- port: SQL port number (depends on server type)
- user: database user
- password: database password
- database: database name
- charset: character set (use latin1 if you get unicode issues)
- encryption: connection type allowed type are NONE, SSL and TLS
- ca_file Trusted CA for SSL and TLS
- cert_file: client certificate file
- key_file: client certificate key

With SQLite, only the 'database' must be set and other parameters are ignored. The database is the full path to an SQLite DB file where WebADM has full write access.

With Oracle, you can optionally use TNS names. If the 'tnsname' is set then the 'host' and 'port' parameters are ignored and a tnsnames.ora file must exist under the conf/ directory.

-->

```
<SqlServer name="SQL Server 1"  
  type="MySQL8"  
  host="webadm1.support.rcdevs.com"  
  user="webadm"  
  charset="utf8mb3"  
  password="webadm"
```

```
password= webadm  
database="webadm"  
encryption="TLS" />
```

```
<SqlServer name="SQL Server 2"  
  type="MySQL8"  
  host="webadm2.support.rcdevs.com"  
  user="webadm"  
  charset="utf8mb3"  
  password="webadm"  
  database="webadm"  
  encryption="TLS"  
 />
```

<!--

A session server is required for storing/sharing persistent memory data on your WebADM server(s). You must specify two servers with clustering. The session server is based on Redis6 which is included in WebADM. With WebADM >= 2.1.5, TLS encryption is used by default on port 4000!

-->

```
<SessionServer name="Session Server 1"  
  host="webadm1.support.rcdevs.com"  
  port="4000"  
  secret="secret" />
```

```
<SessionServer name="Session Server 2"  
  host="webadm2.support.rcdevs.com"  
  port="4000"  
  secret="secret" />
```

<!--

A PKI server (or CA) is required for signing user certificates. The RSign PKI server is included in WebADM. So you can keep the default settings here.

-->

```
<PkiServer name="PKI Server 1"  
  host="webadm1.support.rcdevs.com"  
  port="5000"  
  secret="secret" />
```

```
<PkiServer name="PKI Server 2"  
  host="webadm2.support.rcdevs.com"  
  port="5000"  
  secret="secret" />
```

<!--

HTTP proxy servers can be used by WebADM for connecting remote Web services and version checking.

-->

```
<!--
<ProxyServer name="HTTP Proxy"
  host="proxy"
  port="8080"
  user=""
  password=""
  ca_file="" />
-->







<!--
SMTP mail servers can be used by WebADM for sending emails.
If no server is specified, WebADM will use the local mailer
in /usb/sbin/sendmail to send emails.
-->



















<!--
<MailServer name="SMTP Server"
  host="localhost"
  port="25"
  user=""
  password=""
  encryption="NONE"
  ca_file="" />
-->

</Servers>
```

13.2.3. Download Certificates

WebADM UI: webadm1.support.rcdevs.com

 User Domains (2) Associate domain names with LDAP user search bases.	 Client Policies (0) Define custom policy settings for consumer applications.	 Access Devices (0) Hardware devices for badging and physical access control.
 LDAP Mount Points (1) Connect secondary LDAP servers to the tree view.	 LDAP Option Sets (1) Define LDAP tree constraints for your 'other' administrators.	 Administrator Roles (0) Create admin role templates for your 'other' administrators.

Licensing and Configurations	Runtime Actions
 Software License Details	 Download WebADM CA Certificate
 LDAP Server Details	 Download WebADM SSL Certificate
 LDAP Server Schema	 Issue Server or Client SSL Certificate
 Memory Usage Details	 Clear Admin Session Cache (1 KB) ⓘ
 Hardware Modules Details	 Clear WebADM License Cache ⓘ
 Remote Manager Interface	 Clear WebADM System Caches (297 KB) ⓘ
 Config Object Statuses	 Flush WebADM Session Data (2506 KB) ⓘ
 Network Service Statuses	 Reload WebADM Configurations
 WebADM Base Settings	
 Trusted CA Certificates	

Click on [Download WebADM CA Certificate](#) to download it.

Now click on [Issue Server or Client SSL Certificate](#), add an FQDN: mysqlserver110 and select Server.



User Domains (2)

Associate domain names with LDAP user search bases.



Client Policies (0)

Define custom policy settings for consumer applications.



Access Devices (0)

Hardware devices for badging and physical access control.



LDAP Mount Points (1)

Connect secondary LDAP servers to the tree view.



LDAP Option Sets (1)

Define LDAP tree constraints for your 'other' administrators.











Administrator Roles (0)

Create admin role templates for your 'other' administrators.

Licensing and Configurations

-  [Software License Details](#)
-  [LDAP Server Details](#)
-  [LDAP Server Schema](#)
-  [Memory Usage Details](#)
-  [Hardware Modules Details](#)
-  [Remote Manager Interface](#)
-  [Config Object Statuses](#)
-  [Network Service Statuses](#)
-  [WebADM Base Settings](#)
-  [Trusted CA Certificates](#)

Runtime Actions

-  [Download WebADM CA Certificate](#)
-  [Download WebADM SSL Certificate](#)
-  [Issue Server or Client SSL Certificate](#)
-  [Clear Admin Session Cache \(1 KB\) !\[\]\(3b3789986183f9eec3cad402e3bcf4b9_img.jpg\)](#)
-  [Clear WebADM License Cache !\[\]\(ee8ab7c1986d339c9ebce41d6295b085_img.jpg\)](#)
-  [Clear WebADM System Caches \(297 KB\) !\[\]\(2dc14fcc1a17b84537c735a959a8d154_img.jpg\)](#)
-  [Flush WebADM Session Data \(2506 KB\) !\[\]\(9b441b7646c73c6a8c847d23178e07b5_img.jpg\)](#)
-  [Reload WebADM Configurations](#)

Home Admin Cluster Create Search Import Databases Statistics Applications About Logout

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component.
The certificate is generated with the provided information and signed by WebADM certificate authority.

Auto Confirm Mode

Enable Auto Confirm: Yes No ⓘ

Auto Confirm Time: 5 Minutes ▾

Auto Confirm App: [All] ▾

Auto Confirm IPs: ⓘ

Main information

Server Hostname (FQDN): mysqlserver110

Certificate Type: Server ▾ ⓘ

Certificate validity (in days): 365 ⓘ

Private Key Password (optional): ⓘ

Additional information

Alternative Name(s): ⓘ

Organization Name:

Organizational Unit:

Country Name: ⓘ

Locality Name:

State or Province:

Street Address:

Email Address:

Ok Cancel

Now let's download the `client certificates`:

Click on `Issue Server or Client SSL Certificate`, add an FQDN: mysqlclient110 and select Client.

Home | Admin | Cluster | Create | Search | Import | Databases | Statistics | Applications | About | Logout

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component. The certificate is generated with the provided information and signed by WebADM certificate authority.

Auto Confirm Mode

Enable Auto Confirm: Yes No ⓘ

Auto Confirm Time: 5 Minutes ▾

Auto Confirm App: [All] ▾

Auto Confirm IPs: ⓘ

Main information

Client Name or Description: mysqlclient110

Certificate Type: Client ▾ ⓘ

Restricted Application: [Not Set] ▾ ⓘ

Certificate validity (in days): 365 ⓘ

Private Key Password (optional): ⓘ

Additional information

Organization Name:

Organizational Unit:

Country Name: ⓘ

Locality Name:

State or Province:

Street Address:

Email Address:

Ok Cancel

WebADM UI: webadm2.support.rcdevs.com

For the 2nd Node YOU have to download the certificates by following the same procedure as before:

WebADM CA Certificate

mysqlserver128

mysqlclient128

---Nodes 12---

```
mkdir -p /etc/mysql/ssl/
cd /etc/mysql/ssl/
chown mysql:mysql /etc/mysql/ssl
chown mysql:mysql /etc/mysql/ssl/*
chmod 640 /etc/mysql/ssl/*
```

Copy the Certificates to all the Nodes

```
---Nodes 1---
root@webadm1: cd /etc/mysql/ssl/
root@webadm1: ll
-rw-r----- 1 mysql mysql 1184 Jul  1 08:10 ca.crt
-rw-r--r-- 1 root  root  1476 Jul 11 15:19 mysqlclient110.crt
-rw-r--r-- 1 root  root  1704 Jul 11 15:19 mysqlclient110.key
-rw-r--r-- 1 root  root  1513 Jul 11 15:20 mysqlserver110.crt
-rw-r--r-- 1 root  root  1704 Jul 11 15:20 mysqlserver110.key
```

```
---Nodes 2---
root@webadm2: cd /etc/mysql/ssl/
root@webadm2: ll
-rw-r----- 1 mysql mysql 1184 Jul  1 08:10 ca.crt
-rw-r--r-- 1 root  root  1476 Jul 11 15:19 mysqlclient128.crt
-rw-r--r-- 1 root  root  1704 Jul 11 15:19 mysqlclient128.key
-rw-r--r-- 1 root  root  1513 Jul 11 15:20 mysqlserver128.crt
-rw-r--r-- 1 root  root  1704 Jul 11 15:20 mysqlserver128.key
```

13.2.4. MySQL

Adjust mysqld.cnf

For the section `server_uuid`, you will find this info when you configure [MySQL](#) replication in the coming steps.

```
--Node 1--
root@webadm1: nano /etc/mysql/mysql.conf.d/mysqld.cnf

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid_file    = /var/run/mysqld/mysqld.pid
```

```
# pid-file      = /var/run/mysqld/mysqld.pid
# socket       = /var/run/mysqld/mysqld.sock
# port        = 3306
# datadir     = /var/lib/mysql

server-id      = 1
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 2 because there are 2 Nodes.
auto-increment-increment = 2
# Note: Set "auto-increment-offset" to 1 for Node 1.
auto-increment-offset = 1
replicate-do-db = webadm
log_bin         = mysql-bin
#log-bin        = mysql
#log-basename   = mysql
binlog-do-db   = webadm
log-slave-updates
relay-log      = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
ssl-ca=/etc/mysql/ssl/ca.crt
ssl-cert=/etc/mysql/ssl/mysqlserver110.crt
ssl-key=/etc/mysql/ssl/mysqlserver110.key
require_secure_transport=ON
# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar\_tmpdir
# tmpdir       = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address   = 0.0.0.0
#mysqlx-bind-address = 0.0.0.0
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack       = 256K

# thread_cache_size = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options = BACKUP

# max_connections    = 151

# table_open_cache   = 4000
#
# * Logging and Replication
#
```

```
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
general_log_file      = /var/log/mysql/query.log
general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
# slow_query_log      = 1
# slow_query_log_file = /var/log/mysql/mysql-slow.log
# long_query_time = 2
# log-queries-not-using-indexes
#

[client]
ssl-ca=/etc/mysql/ssl/ca.crt
ssl-cert=/etc/mysql/ssl/mysqlclient110.crt
ssl-key=/etc/mysql/ssl/mysqlclient110.key

[auto]
server_uuid=635bcc36-f868-11ec-9e7b-000c2921c70f
```

```
--Node 2--
root@webadm1: nano /etc/mysql/mysql.conf.d/mysqld.cnf

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user      = mysql
# pid-file = /var/run/mysqld/mysqld.pid
```

```
# socket      = /var/run/mysqld/mysqld.sock
# port       = 3306
# datadir    = /var/lib/mysql

server-id     = 2
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 2 because there are 2 Nodes.
auto-increment-increment = 2
# Note: Set "auto-increment-offset" to 2 for Node 2.
auto-increment-offset = 2
replicate-do-db = webadm
log_bin         = mysql-bin
#log-bin        = mysql
#log-basename   = mysql
binlog-do-db   = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
ssl-ca=/etc/mysql/ssl/ca.crt
ssl-cert=/etc/mysql/ssl/mysqlserver128.crt
ssl-key=/etc/mysql/ssl/mysqlserver128.key
require_secure_transport=ON
# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar\_tmpdir
# tmpdir       = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address   = 0.0.0.0
#mysqlx-bind-address = 0.0.0.0
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack       = 256K

# thread_cache_size = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options = BACKUP

# max_connections    = 151

# table_open_cache   = 4000
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
```



```

#
# Log all queries
# Be aware that this log type is a performance killer.
general_log_file      = /var/log/mysql/query.log
general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
# slow_query_log       = 1
# slow_query_log_file = /var/log/mysql/mysql-slow.log
# long_query_time = 2
# log-queries-not-using-indexes
#

[client]
ssl-ca=/etc/mysql/ssl/ca.crt
ssl-cert=/etc/mysql/ssl/mysqlclient128.crt
ssl-key=/etc/mysql/ssl/mysqlclient128.key

[auto]
server_uuid=635bcc36-f868-11ec-9e7b-000c2921c70f

```

Configure TLS Replication:

```
---NODES 12---
```

```
root@webadm: mysql -u root -p
```

```
mysql> SET sql_log_bin = 0;
```

```
mysql> CREATE DATABASE webadm;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> CREATE USER 'webadm'@'webadm1.support.rcdevs.com' IDENTIFIED BY 'webadm' REQUIRE SSL;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CREATE USER 'webadm'@'webadm2.support.rcdevs.com' IDENTIFIED BY 'webadm' REQUIRE SSL;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON *.* to 'root'@'webadm1.support.rcdevs.com' IDENTIFIED BY
'password';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION_APPLIER ON *.* TO 'webadm'@'webadm1.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION_APPLIER ON *.* TO 'webadm'@'webadm2.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> ALTER USER 'webadm'@'webadm1.support.rcdevs.com' REQUIRE SSL;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> ALTER USER 'webadm'@'webadm2.support.rcdevs.com' REQUIRE SSL;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm1.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm2.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm1.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm2.support.rcdevs.com';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SET sql_log_bin = 1;
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> STOP SLAVE;
```

```
mysql> SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000007 | 13010 | webadm      |                   |                   |
+-----+-----+-----+-----+-----+
1 row in set (0.03 sec)
```

Let's start with the **NODE 2** and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from **NODE 1**.

```
---NODES 2---
```

```
mysql> CHANGE MASTER TO MASTER_HOST = 'webadm1.support.rcdevs.com', MASTER_USER =
'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mysql-bin.000007',
MASTER_LOG_POS = 13010, MASTER_SSL=1;
```

```
Query OK, 0 rows affected (0.05 sec)
```

Continue with the **NODE 1** and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from **NODE 2**.

```
---NODES 1---
mysql> CHANGE MASTER TO MASTER_HOST = 'webadm2.support.rcdevs.com', MASTER_USER =
'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mysql-bin.000001',
MASTER_LOG_POS = 13043, MASTER_SSL=1;
```

Query OK, 0 rows affected (0.06 sec)

```
---NODES12---
```

```
mysql> START SLAVE;
```

13.2.5. Verify Replication Status

```
---NODES 1---
```

```
mysql> SHOW SLAVE STATUS \G
```

```
***** 1. row *****
Slave_IO_State: Waiting for source to send event
Master_Host: webadm2.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000008
Read_Master_Log_Pos: 12806
Relay_Log_File: slave-relay.000014
Relay_Log_Pos: 373
Relay_Master_Log_File: mysql-bin.000008
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 12806
Relay_Log_Space: 748
Until_Condition: None
Until_Log_File:
```

```

    Until_Log_Pos: 0
    Master_SSL_Allowed: Yes
    Master_SSL_CA_File:
    Master_SSL_CA_Path:
    Master_SSL_Cert:
    Master_SSL_Cipher:
    Master_SSL_Key:
    Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
    Last_IO_Errno: 0
    Last_IO_Error:
    Last_SQL_Errno: 0
    Last_SQL_Error:
Replicate_Ignore_Server_Ids:
    Master_Server_Id: 2
        Master_UUID: 635bcc36-f868-11ec-9e7b-000c2921c70f
    Master_Info_File: mysql.slave_master_info
    SQL_Delay: 0
    SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Replica has read all relay log; waiting for more updates
    Master_Retry_Count: 86400
    Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
    Master_SSL_Crl:
    Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
    Auto_Position: 0
Replicate_Rewrite_DB:
    Channel_Name:
    Master_TLS_Version:
Master_public_key_path:
    Get_master_public_key: 0
    Network_Namespace:
1 row in set, 1 warning (0.04 sec)

```

```
mysql> SHOW VARIABLES LIKE '%ssl%';
```

```

+-----+-----+
| Variable_name          | Value          |
+-----+-----+
| admin_ssl_ca           |                |
| admin_ssl_capath       |                |
| admin_ssl_cert         |                |
| admin_ssl_cipher       |                |
| admin_ssl_crl          |                |
| admin_ssl_crlpath      |                |
| admin_ssl_key          |                |

```

```

| have_openssl          | YES          |
| have_ssl              | YES          |
| mysqlx_ssl_ca         |              |
| mysqlx_ssl_capath    |              |
| mysqlx_ssl_cert       |              |
| mysqlx_ssl_cipher     |              |
| mysqlx_ssl_crl        |              |
| mysqlx_ssl_crlpath    |              |
| mysqlx_ssl_key        |              |
| performance_schema_show_processlist | OFF          |
| ssl_ca                | /etc/mysql/ssl/ca.crt |
| ssl_capath            |              |
| ssl_cert              | /etc/mysql/ssl/mysqlserver110.crt |
| ssl_cipher            |              |
| ssl_crl               |              |
| ssl_crlpath           |              |
| ssl_fips_mode         | OFF          |
| ssl_key               | /etc/mysql/ssl/mysqlserver110.key |
| ssl_session_cache_mode | ON           |
| ssl_session_cache_timeout | 300         |
+-----+-----+

```

27 rows in set (1.68 sec)

mysql> status;

mysql Ver 8.0.29-0ubuntu0.22.04.2 for Linux on x86_64 ((Ubuntu))

Connection id: 15821

Current database:

Current user: root@localhost

SSL: Cipher in use is TLS_AES_256_GCM_SHA384

Current pager: stdout

Using outfile: ''

Using delimiter: ;

Server version: 8.0.29-0ubuntu0.22.04.2 (Ubuntu)

Protocol version: 10

Connection: Localhost via UNIX socket

Server character set: utf8mb4

Db character set: utf8mb4

Client character set: utf8mb4

Conn. character set: utf8mb4

UNIX socket: /var/run/mysqld/mysqld.sock

Binary data as: Hexadecimal

Uptime: 22 hours 6 min 59 sec

Threads: 10 Questions: 63781 Slow queries: 0 Opens: 229 Flush tables: 3 Open tables: 148 Queries per second avg: 0.801

```
mysql> exit
Bye
```

```
---NODES 2---
```

```
mysql> SHOW SLAVE STATUS \G
```

```
***** 1. row *****
Slave_IO_State: Waiting for source to send event
Master_Host: webadm1.support.rcdevs.com
Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000011
Read_Master_Log_Pos: 13010
Relay_Log_File: slave-relay.000015
Relay_Log_Pos: 13226
Relay_Master_Log_File: mysql-bin.000011
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 13010
Relay_Log_Space: 13601
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: Yes
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID: 15f82f6c-f60c-11e0-9764-000c202a0246
```

Master_UUID: 1316a10c-10ee-11ec-9764-000c292e0240

Master_Info_File: mysql.slave_master_info

SQL_Delay: 0

SQL_Remaining_Delay: NULL

Slave_SQL_Running_State: Replica has read all relay log; waiting for more updates

Master_Retry_Count: 86400

Master_Bind:

Last_IO_Error_Timestamp:

Last_SQL_Error_Timestamp:

Master_SSL_Crl:

Master_SSL_Crlpath:

Retrieved_Gtid_Set:

Executed_Gtid_Set:

Auto_Position: 0

Replicate_Rewrite_DB:

Channel_Name:

Master_TLS_Version:

Master_public_key_path:

Get_master_public_key: 0

Network_Namespace:

1 row in set, 1 warning (0.21 sec)

mysql> SHOW VARIABLES LIKE '%ssl%';

Variable_name	Value
admin_ssl_ca	
admin_ssl_capath	
admin_ssl_cert	
admin_ssl_cipher	
admin_ssl_crl	
admin_ssl_crlpath	
admin_ssl_key	
have_openssl	YES
have_ssl	YES
mysqlx_ssl_ca	
mysqlx_ssl_capath	
mysqlx_ssl_cert	
mysqlx_ssl_cipher	
mysqlx_ssl_crl	
mysqlx_ssl_crlpath	
mysqlx_ssl_key	
performance_schema_show_processlist	OFF
ssl_ca	/etc/mysql/ssl/ca.crt
ssl_capath	
ssl_cert	/etc/mysql/ssl/mysqlserver128.crt
ssl_cipher	
ssl_crl	
ssl_crlpath	
ssl_fips_mode	OFF

```

| ssl_key                | /etc/mysql/ssl/mysqlserver128.key |
| ssl_session_cache_mode | ON                                |
| ssl_session_cache_timeout | 300                              |
+-----+-----+
27 rows in set (0.67 sec)

mysql> status;
-----
mysql Ver 8.0.29-0ubuntu0.22.04.2 for Linux on x86_64 ((Ubuntu))

Connection id: 15739
Current database:
Current user: root@localhost
SSL: Cipher in use is TLS_AES_256_GCM_SHA384
Current pager: stdout
Using outfile: ''
Using delimiter: ;
Server version: 8.0.29-0ubuntu0.22.04.2 (Ubuntu)
Protocol version: 10
Connection: Localhost via UNIX socket
Server characterset: utf8mb4
Db characterset: utf8mb4
Client characterset: utf8mb4
Conn. characterset: utf8mb4
UNIX socket: /var/run/mysqld/mysqld.sock
Binary data as: Hexadecimal
Uptime: 22 hours 7 min 9 sec

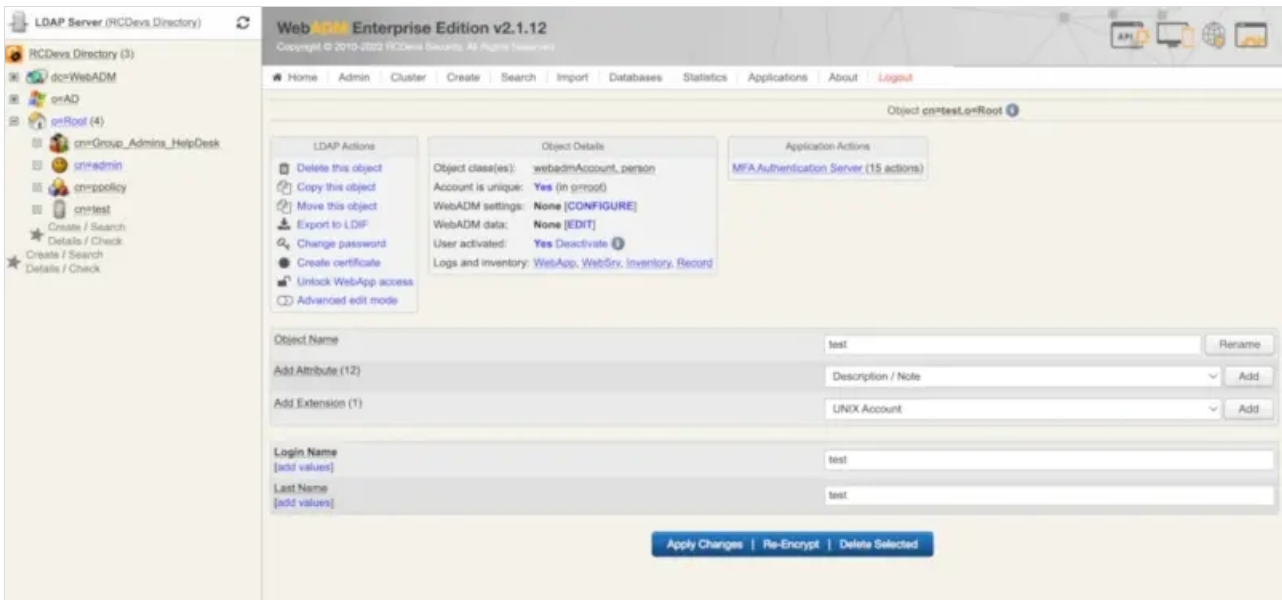
Threads: 9 Questions: 63246 Slow queries: 0 Opens: 164 Flush tables: 3 Open tables: 83 Queries per
second avg: 0.794
-----

mysql> exit
Bye

```

13.2.6. Test of replication

To test our replication we are going to make a modification in the first Node and see if it replicates in the second one. For example, we will create a user in the first server:



We will check if the creation of the new user is replicated on the second server:

---NODES 2---

```
mysql> use webadm;
```

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

```
mysql> select * from Admin;
```

```

+----+-----+-----+-----+-----+-----+
--+
| ID | Time          | DN          | Source    | Session | Text |
+----+-----+-----+-----+-----+-----+
--+
| 441 | 2022-07-12 13:40:13 | cn=admin,o=Root | 192.168.45.163 | XORE08JC | Created object 'cn=test,o=Root'(webadmaccount, person, inetorgperson)
+----+-----+-----+-----+-----+-----+
--+
```

We can see the modification on the second server, it means that the **replication** works. You will find the new user created in WebADM UI as well.

13.3. MySQL8 replication without TLS

Tested on **CentOS Stream 9**

After installing **MySQL8** we will configure the **replication**:

13.3.1. Adjust server.cnf

For the section `server_uuid`, you will find this info when you configure `MySQL replication` in the coming steps.

```
--Node 1--
root@webadm1: nano /etc/my.cnf.d/mysql-server.cnf

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid-file    = /var/run/mysqld/mysqld.pid
# socket      = /var/run/mysqld/mysqld.sock
# port        = 3306
# datadir     = /var/lib/mysql

server-id     = 1
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 2 because there are 2 Nodes.
auto-increment-increment = 2
# Note: Set "auto-increment-offset" to 1 for Node 1.
auto-increment-offset = 1
replicate-do-db = webadm
log_bin        = mysql-bin
#log-bin       = mysql
#log-basename  = mysql
binlog-do-db   = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir       = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
```

```
bind-address          = 0.0.0.0
#mysqlx-bind-address = 0.0.0.0
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack       = 256K

# thread_cache_size  = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options = BACKUP

# max_connections    = 151

# table_open_cache   = 4000
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
general_log_file      = /var/log/mysql/query.log
general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# binlog_expire_logs_seconds = 2592000
max_binlog_size = 100M
# Here you can see queries with especially long duration
# slow_query_log          = 1
# slow_query_log_file    = /var/log/mysql/mysql-slow.log
# long_query_time = 2
# log-queries-not-using-indexes

[auto]
server_uuid=32d53449-0387-11ed-8bf7-000c29f8172c
```

```
--Node 2--
root@webadm1: nano /etc/my.cnf.d/mysql-server.cnf

#
# The MySQL database server configuration file.
```

```

#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid-file    = /var/run/mysqld/mysqld.pid
# socket      = /var/run/mysqld/mysqld.sock
# port        = 3306
# datadir     = /var/lib/mysql

server-id      = 2
replicate-same-server-id = 0
# Note: Set "auto-increment-increment" to 2 because there are 2 Nodes.
auto-increment-increment = 2
# Note: Set "auto-increment-offset" to 2 for Node 2.
auto-increment-offset = 2
replicate-do-db = webadm
log_bin         = mysql-bin
#log-bin        = mysql
#log-basename   = mysql
binlog-do-db   = webadm
log-slave-updates
relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir       = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address      = 0.0.0.0
#mysqlx-bind-address = 0.0.0.0
#
# * Fine Tuning
#
key_buffer_size   = 16M
# max_allowed_packet = 64M
# thread_stack    = 256K

```

```

# thread_stack          = 250K

# thread_cache_size     = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
mysam-recover-options = BACKUP

# max_connections       = 151

# table_open_cache      = 4000
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
general_log_file        = /var/log/mysql/query.log
general_log             = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# binlog_expire_logs_seconds = 2592000
max_binlog_size = 100M
# Here you can see queries with especially long duration
# slow_query_log         = 1
# slow_query_log_file    = /var/log/mysql/mysql-slow.log
# long_query_time = 2
# log-queries-not-using-indexes
#

[auto]
server_uuid=2eb91828-0387-11ed-8a59-000c29e7d978

```

13.3.2. Configure Replication

```

---NODES 12---

root@webadm: mysql -u root -p

mysql> SET sql_log_bin = 0;

mysql> CREATE DATABASE webadm;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE USER 'webadm'@'webadm1.support.rcdevs.com' IDENTIFIED WITH

```

```
mysql_native_password BY 'webadm';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CREATE USER 'webadm'@'webadm2.support.rcdevs.com' IDENTIFIED WITH  
mysql_native_password BY 'webadm';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION_APPLIER ON *.* TO 'webadm'@'webadm1.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION_APPLIER ON *.* TO 'webadm'@'webadm2.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> ALTER USER 'webadm'@'webadm1.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> ALTER USER 'webadm'@'webadm2.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm1.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON webadm.* to 'webadm'@'webadm2.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm1.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'webadm'@'webadm2.support.rcdevs.com';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SET sql_log_bin = 1;
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> STOP SLAVE;
```

```
mysql> SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+-----+  
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |  
+-----+-----+-----+-----+-----+  
| mysql-bin.000007 | 13010 | webadm      |                   |                   |  
+-----+-----+-----+-----+-----+  
1 row in set (0.03 sec)
```

Let's start with the **NODE 2** and replace the `MASTER_LOG_FILE` name and the `MASTER_LOG_POS` number with the values of `SHOW MASTER STATUS` from **NODE 1**.

---NODES 2---

```
mysql> CHANGE MASTER TO MASTER_HOST = 'webadm1.support.rcdevs.com', MASTER_USER =  
'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mysql-bin.000007', MASTER_LOG_POS  
= 13010;
```

Query OK, 0 rows affected (0.05 sec)

Continue with the **NODE 1** and replace the **MASTER_LOG_FILE** name and the **MASTER_LOG_POS** number with the values of **SHOW MASTER STATUS** from **NODE 2**.

---NODES 1---

```
mysql> CHANGE MASTER TO MASTER_HOST = 'webadm2.support.rcdevs.com', MASTER_USER =  
'webadm', MASTER_PASSWORD = 'webadm', MASTER_LOG_FILE = 'mysql-bin.000001',  
MASTER_LOG_POS = 13043;
```

Query OK, 0 rows affected (0.06 sec)

---NODES1---

```
mysql> START SLAVE;
```

```
mysql> SHOW SLAVE STATUS \G
```

```
mysql> SHOW SLAVE STATUS \G
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for source to send event
```

```
Master_Host: webadm2.support.rcdevs.com
```

```
Master_User: webadm
```

```
Master_Port: 3306
```

```
Connect_Retry: 60
```

```
Master_Log_File: mysql-bin.000005
```

```
Read_Master_Log_Pos: 9800
```

```
Relay_Log_File: slave-relay.000018
```

```
Relay_Log_Pos: 326
```

```
Relay_Master_Log_File: mysql-bin.000005
```

```
Slave_IO_Running: Yes
```

```
Slave_SQL_Running: Yes
```

```
Replicate_Do_DB: webadm
```

```
Replicate_Ignore_DB:
```

```
Replicate_Do_Table:
```

```
Replicate_Ignore_Table:
```

```
Replicate_Wild_Do_Table:
```

```
Replicate_Wild_Ignore_Table:
```

```
Last_Errno: 0
```

```
Last_Error:
```

```
Skip_Counter: 0
```

```
Exec_Master_Log_Pos: 9800
```

```
EXEC_Master_Log_Pos: 9000
Relay_Log_Space: 881
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 2
Master_UUID: 32d53449-0387-11ed-8bf7-000c29f8172c
Master_Info_File: mysql.slave_master_info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Replica has read all relay log; waiting for more updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
Master_public_key_path:
Get_master_public_key: 0
Network_Namespace:
1 row in set, 1 warning (0.02 sec)
```

---NODES1---

```
mysql> START SLAVE;
mysql> SHOW SLAVE STATUS \G
```

```
***** 1. row *****
Slave_IO_State: Waiting for source to send event
Master Host: webadm1.support.rcdevs.com
```


Master_User: webadm
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000006
Read_Master_Log_Pos: 1787
Relay_Log_File: slave-relay.000018
Relay_Log_Pos: 2003
Relay_Master_Log_File: mysql-bin.000006
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: webadm
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1787
Relay_Log_Space: 2378
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID: 2eb91828-0387-11ed-8a59-000c29e7d978
Master_Info_File: mysql.slave_master_info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Replica has read all relay log; waiting for more updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:

```
Retrieved_Gtid_Set:
Executed_Gtid_Set:
  Auto_Position: 0
Replicate_Rewrite_DB:
  Channel_Name:
  Master_TLS_Version:
Master_public_key_path:
Get_master_public_key: 0
  Network_Namespace:
1 row in set, 1 warning (0.01 sec)
```

13.3.3. Verify Replication Status

14. Register Web Applications and Web Services

Once your graphical setup is done, you can register and configure the different applications and services you installed. Login on WebADM Admin Portal and click on **Applications** tab.

To do it, just click on the **REGISTER** button for the Application/Service you want.

The screenshot shows the WebADM Enterprise Edition v2.1.10 Admin Portal. The main content area is titled "Registered Applications and Services" and displays a list of web services. On the left, there is a "Categories" sidebar with "Authentication (2)" selected. The main list shows two services:

- MFA Authentication Server (OpenOTP) v2.1.2 (Commercial)**: Multi-factor authentication service supporting OATH HOTP/TOTP/OCRA, FIDO, YubiKey, SMS OTP and Mail OTP. Latest Version: 2.1.2 (Ok). Status: Not Registered [REGISTER]. Service URL (SSL): https://192.168.4.20:8443/openotp/. Service URL (STD): http://192.168.4.20:8080/openotp/. Mobile Endpoint: https://192.168.4.20/ws/openotp/. U2F Facet Endpoint: https://192.168.4.20/ws/appid/. SOAP WSDL File: openotp.wsdl.
- SSH Public Key Server (SpanKey) v2.0.19 (Commercial)**: SSH public key distribution service supporting RSA, DSA and ECC keys. Latest Version: 2.0.19 (Ok). Status: Not Registered [REGISTER]. Service URL (SSL): https://192.168.4.20:8443/spankey/. Service URL (STD): http://192.168.4.20:8080/spankey/. U2F Application ID: https://192.168.4.20/ws/appid-ssh/. SOAP WSDL File: spankey.wsdl.

The status of the application will switch to **Enabled**.


WebADM Enterprise Edition v2.1.10
Copyright © 2010-2022 RCDevs Security, All Rights Reserved

Home | Admin | Create | Search | Import | Databases | Statistics | Applications | About | Logout

Registered Applications and Services

Categories
✓ Authentication (2)
SMS Relay (1)
Self-Service (3)
Single Sign-On (2)

Web Services

 **MFA Authentication Server (OpenOTP) v2.1.2 (Commercial)**

Multi-factor authentication service supporting OATH HOTP/TOTP/OCRA, FIDO, YubiKey, SMS OTP and Mail OTP.

Latest Version: 2.1.2 (Ok)

Status: **Enabled** [CONFIGURE] [REMOVE]

Service URL (SSL): <https://192.168.4.20:8443/openotp/>

Service URL (STD): <http://192.168.4.20:8080/openotp/>

Mobile Endpoint: <https://192.168.4.20/ws/openotp/>

U2F Facet Endpoint: <https://192.168.4.20/ws/appid/>

SOAP WSDL File: [openotp.wsdl](#)

You can now click **CONFIGURE** button to start the configuration.

Installation of your Standalone WebADM/Cluster is done, you can continue with the [WebADM Administrator Guide](#).

This manual was prepared with great care. However, RCDevs Security S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs Security S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs Security S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs Security S.A. The latter especially applies for data processing systems. RCDevs Security S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs Security. © 2024 RCDevs Security S.A., All Rights Reserved