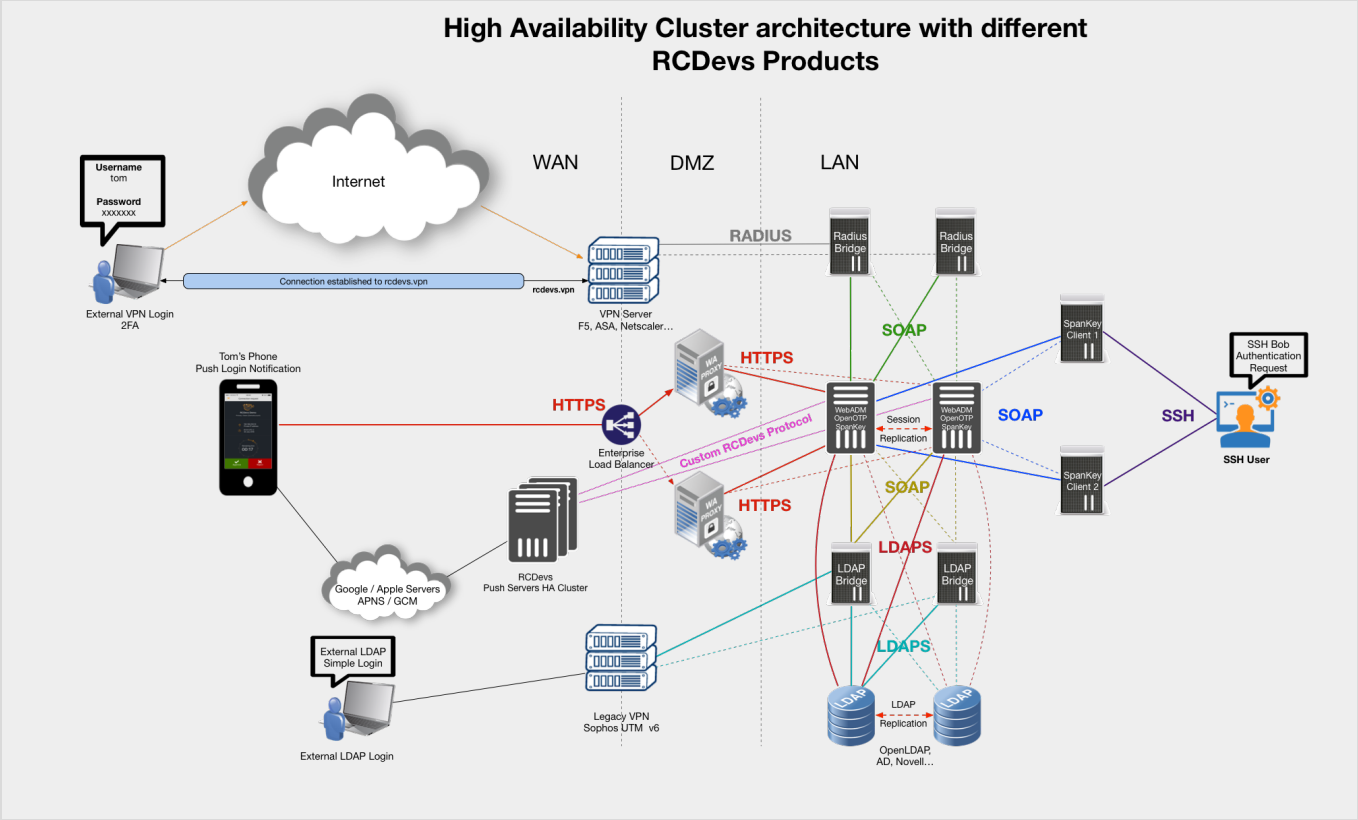
# WEBADM HIGH AVAILABILITY GUIDE

# WebADM High Availability Guide

HA Cluster




## 1. Product Documentation

This document is a deployment guide for RCDevs WebADM in high availability (or cluster) mode. The reader should notice that this document is not a guide for installing WebADM applications (Web Services and WebApps).

## 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 container for RCDevs Web Services and WebApps such as OpenOTP. WebADM requires an LDAP directory as back-end user store and a SQL database for logs and end-user message customizations. WebADM is compatible with Novell eDirectory, OpenLDAP, RCDevs Directory Server and Microsoft ActiveDirectory $^{2003}/_{2008}$.

## 3. System Requirements

The current version of WebADM runs on Linux 32bit or 64bit operating systems with GLIBC >= 2.5. The installation package contains all the required dependencies allowing WebADM to run on any Linux-based system without other requirements. WebADM only needs an LDAP backend (Novell eDirectory, OpenLDAP, RCDevs Directory Server, Microsoft ActiveDirectory, or Oracle Directory) and a SQL backend (MySQL, PostgreSQL, Oracle or Microsoft SQL).

For running WebADM and its applications, as well as the OpenOTP Radius Bridge server and RCDevs Directory Server, your system should fit the following requirements:

- A dedicated server computer or Virtual machine with Linux GLIBC >= 2.5 (RedHat, Centos, SuSe Debian, Ubuntu).
- 2 GHz processor (multi-core / multi-thread processor is highly recommended). Both 32 and 64-bit chips are supported provided that 32 libraries are present.
- 2GB RAM memory.
- 2GB disk space for installation files.
- Network access with DNS and NTP integration.
- A local or remote LDAP directory server (RCDevs Directory Server, OpenLDAP, Novell eDirectory or Microsoft ActiveDirectory >= 2003). WebADM for ActiveDirectory 2003 has some limitations which do not exist with ActiveDirectory 2008. Always prefer using ActiveDirectory 2008 with WebADM.
- A local or remote SQL database server (MySQL, PostgreSQL). Oracle and MS SQL Server support are included but setup might require manual table creation.
- Outbound Internet access for checking versions, connecting SMS gateways and sending emails.
- A local mail transfer agent (Sendmail or Postfix).
- Firewall open ports: 80, 443, 8080, 8443, 1812. Some other ports are required for cluster node communications as described later.

# 4. High Availability Mechanisms

**⚠ Warning**

Starting from WebADM version 1.4.2, any high availability and clustering feature require an RCDevs Enterprise license. Without a valid license file, the HA and cluster features are automatically disabled.

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.

## 4.1 Connecting Redundant External Services

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 will re-check if the primary service has recovered every minute. 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).

### 4.1.1 Connecting Two 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 an OpenLDAP require LDAP replication configuration. Please refer to the OpenLDAP documentation for OpenLDAP replication.

> **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 server (Novell eDirectory, OpenLDAP or RCDevs directory server) can be installed and run on one or several of the cluster nodes. They can be deployed on another dedicated server too.

### 4.1.2 Connecting Two 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" />
```

It is preferred that SQL databases use replication but this is not a requirement. It's a requirement if you use Hardware Tokens with WebADM because Token inventory is stored in the SQL.

## 4.2 Installing WebADM In Cluster-Mode

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.

### 4.2.1 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 and SOAP 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 one central certificate authority. And only one cluster node will play 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**

This component handles all the user sessions initiated by the 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 ensures 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 communicated internal information too, such as configuration updates.

> **Note**
>
> With WebADM >= 1.2.6-1, the session manager supports automatic synchronous replication. Session data are replicated in real-time between the two first session servers in your configuration. Failover to the secondary node does also not break running sessions.

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**

One node is assigned the certificate authority role. 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.

## 5. Cluster Setup

In this section, we will describe how to set up a cluster configuration for WebADM and Radius Bridge. The cluster will provide redundant web services (ex. OpenOTP), WebApps and RADIUS authentication services.

### 5.1 Installing The First Node

The first node of your cluster is a standard WebADM installation and there is nothing specific to be configured on your first WebADM system. Yet, some firewall ports will have to be opened for allowing the others nodes to communicate with the internal services such as the session manager and PKI server.

The setup of the primary node is started with the command `/opt/webadm/bin/setup`. The setup will initiate the CA, create local service certificates, setup permissions, etc… The configuration of the `servers.xml` file will contain the following information:

LDAP Servers:

- LDAP 1
- LDAP 2

SQL Servers:

- SQL 1

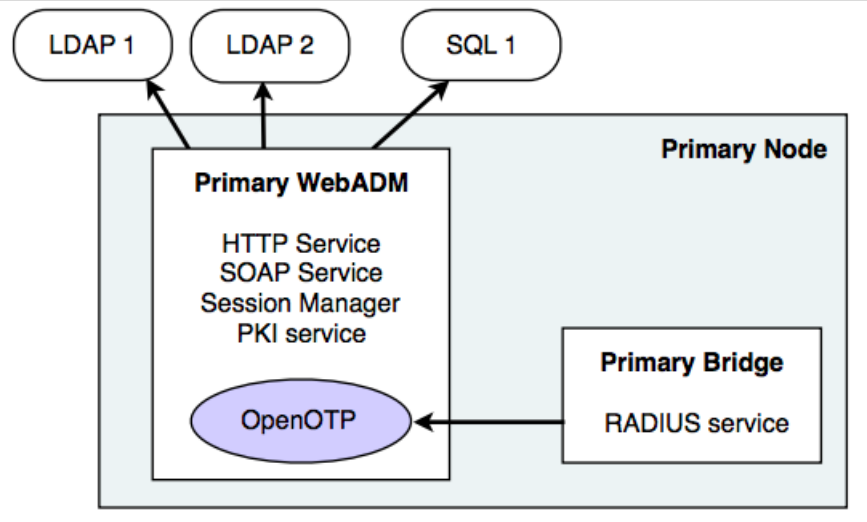
Session Manager:

- Localhost

› <secondary server>

PKI Server:

› Localhost




In this example, we connect two LDAP servers for redundancy and only one SQL server. The Radius Bridge is installed on the same server running WebADM and uses the following OpenOTP URL in the `/opt/radiusd/conf/openotp.conf`: http://localhost:8080/openotp/

## 5.2 Installing A Secondary Node

The node is installed with the self-installer packages like with the primary node but the setup script must be run using the `slave` parameter with the command: `/opt/webadm/bin/setup slave`.

The setup can be re-run on an existing installation. You can also install a second VMWare appliance and re-run its WebADM setup script after installation for adding the node to your cluster.

The secondary node should use the same configuration files as the primary node. You can copy the `/opt/webadm/conf/webadm.conf` file from the primary node. Special attention should be given to the LDAP encryption key which must be the same on all your cluster nodes.

The `/opt/webadm/conf/servers.xml` should use the same LDAP / SQL servers and in the same order. The session management services will be running on both servers but only one of them must be used at a time by both servers. The two session managers can also be specified in the `servers.xml` files, but in the same order. It is possible to use the local session manager on both servers when both WebADM servers are used in failover only and are never used at the same time.

The PKI server will not be set up nor run on the secondary server. The server will use the primary server PKI. During the setup in slave mode, the script will ask for the IP address, port number and secret of the primary server PKI. It will communicate with the remote PKI to initialize its SSL certificates ad CA trusts.

Proceed with the following steps for your secondary node installation:

1) On the primary server, allow client PKI connections to the Rsignd PKI server. This is done by adding a client configuration block

for the secondary server in the `/opt/webadm/conf/rsignd.conf` file:

```
client {
 hostname 127.0.0.1
 secret secret
 services getcacert signcsr
}

client {
 hostname <secondary node IP>
 secret secret
 services getcacert signcsr
 }
```

You can add the secondary server's session manager in the `/opt/webadm/conf/servers.xml` for session manager redundancy:

```
<SessionServer name="Session Server 1"
 host="localhost"
 port="4000" />
<SessionServer name="Session Server 2"
 host="secondary node IP"
 port="4000" />
<PkiServer name="PKI Server"
 host="localhost"
 port="5000"
 secret="secret" />
```

Restart the WebADM server with the command: /opt/webadm/bin/webadm restart.

2) On the primary server, you must allow network communication to the session manager and PKI server ports from the secondary server. On Linux edit the /etc/sysconfig/iptables file and the line:

```
# Port for PKI server
-A INPUT -p tcp -m tcp -s <secondary node IP> -j ACCEPT --dport 5000
# Port for Session Manager access & session replications (for WebADM >= 1.3.x)
-A INPUT -p tcp -m tcp -m multiport -s <secondary node IP> -j ACCEPT --dports
11211,11212
-A INPUT -p udp -m udp -m multiport -s <secondary node IP> -j ACCEPT --dports
11211,11212
# Port for Session Manager access & session replications (for WebADM 1.4.x)
-A INPUT -p tcp -m tcp -s <secondary node IP> -j ACCEPT --dport 4000
Port TCP 5000 is used for the PKI server.
Port TCP 11211 is used for the session manager on WebADM 1.3.x.
Port TCP 4000 is used for the session manager on WebADM >= 1.4.x.
```

Also add a firewall rule for SOAP services inter-communications:

```
-A INPUT -p tcp -m tcp -m multiport -s <secondary node IP> -j ACCEPT --dports 8080,8443
```

Restart the local firewall with the command:

```
/etc/init.d/iptables restart
```

3) On the secondary server, run the setup script in slave mode with the command:

```
/opt/webadm/bin/setup slave
```

You will be asked for the PKI server IP address, port, secret. The address is the primary node IP. The port is 5000. And the secret is 'secret' or the secret you have defined in the /opt/webadm/conf/rsignd.conf file on the primary server for the secondary server client. The SSL certificates are generated on the primary node and the CA certificate is installed in the local CA trust list.

4) On the secondary server, configure the `/opt/webadm/conf/servers.xml` file to use the session manager and PKI server from the primary server.

```
<SessionServer name="Session Server 1"
 host="primary node IP”
 port="4000" />

<SessionServer name="Session Server 2"
 host="localhost"
 port="4000" />

<PkiServer name="PKI Server"
 host="primary node IP"
 port="5000"
 secret="secret" />
```

**⚠ Warning**

Note here that the first declared session manager is the primary server. And there is no PKI server redundancy.

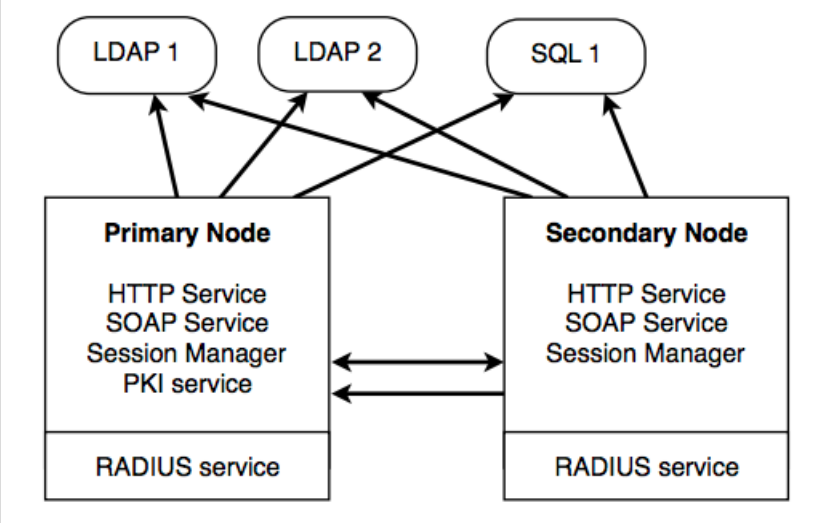
5) On the secondary server, add the firewall rules to allow communications from the primary server.

```
# Ports for Session Manager access & session replications
-A INPUT -p tcp -m tcp -s <primary node IP> -j ACCEPT --dport 4000
# Ports for WebADM SOAP server
-A INPUT -p tcp -m tcp -m multiport -s <primary node IP> -j ACCEPT --dports 8080,8443
```

You can now start the WebADM server on the secondary node. IMPORTANT: If you get a message like “Connected Session server: ERROR (no servers available)” when starting the WebADM server, then be sure the TCP port 4000 is correctly opened in both directions (on both server for the other node IP). You can do the following command to check if the remote port is opened.

```
telnet <Other Node IP> 4000
```

6) On the secondary node, configure the Radius Bridge exactly like on the primary node. You cluster configuration will look like this:




## 5.3 LDAP Replication

LDAP replication may differ according to the chosen LDAP implementation. With Active Directory, replication is handled by the Domain Controllers. With Novell eDirectory, replication requires a partition to be set up and replication should be configured with Novell eManager. With RCDevs Directory Server and more generally with OpenLDAP, the replication uses the syncprov overlay. The recommended is a master-master mirror configuration. On the master node, edit the `/opt/slapd/conf/slapd.conf` file, uncomment the replication block and configure it this way:

```
serverID 1
syncrepl rid=001
 provider=ldap://<secondary node IP>
 bindmethod=simple
 binddn="cn=admin,o=Root"
 credentials="your admin password"
 starttls=yes
 tls_reqcert=never
 searchbase=""
 schemachecking=on
 type=refreshAndPersist
 retry="60 +"
mirrormode on
```

On the secondary node, configure the replication this way:

```
serverID 2
syncrepl rid=001
 provider=ldap://<primary node IP>
 bindmethod=simple
 binddn="cn=admin,o=Root"
 credentials="your admin password"
 starttls=yes
 tls_reqcert=never
 searchbase=""
 schemachecking=on
 type=refreshAndPersist
 retry="60 +"
mirrormode on
```

On both node, be sure to authorize the LDAP port at the firewall level by adding the rules below:

On primary node:

```
-A INPUT -p tcp -m tcp -s <secondary node IP> -j ACCEPT --dport 389
```

On secondary node:

```
-A INPUT -p tcp -m tcp -s <primary node IP> -j ACCEPT --dport 389
```

## 6. Common Cluster Scenarios

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.

## 6.1 Load-balanced + Failover WebADM Cluster

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.

The PKI server runs on the primary WebADM server only. The second server is configured to contact Server 1 for any PKI operation. This is a requirement in any cluster installation since there can be only one certificate authority on the cluster. Note that having the PKI service down does not impact the normal operations of the cluster.

**On Server 1**

```
LDAP Servers: LDAP 1, LDAP 2
SQL Servers: SQL 1, SQL 2
Session Manager: Localhost, Server 2
PKI Server: Localhost
```

**On Server 2**

```
LDAP Servers: LDAP 1, LDAP 2
SQL Servers: SQL 1, SQL 2
Session Manager: Server 1, Localhost
PKI Server: Server 1
```

## 6.2 Failover WebADM Cluster

In this mode, only the primary WebADM server is used in a normal situation. The secondary server ensures redundancy and is used only in the event where the primary server is not available.

The remote services (LDAP servers and SQL servers) can be used in the same order or in a different order. This is not important since the two cluster nodes are not used at the same time and do not require a real-time LDAP data consistency. With clusters having the LDAP services deployed directly on the cluster nodes (Ex. RCDevs Directory Server), both servers may be connected to their local LDAP only.

Both servers can use their local session manager only as they do not need to share sessions and distributed locks.

The PKI server still needs to be run on the primary WebADM server only (for the same reasons as explained previously).

**On Server 1**

```
LDAP Servers: LDAP 1, LDAP 2
SQL Servers: SQL 1, SQL 2
Session Manager: Localhost
PKI Server: Localhost
```

**On Server 2 (Alternative 1)**

```
LDAP Servers: LDAP 1, LDAP 2
SQL Servers: SQL 1, SQL 2
Session Manager: Localhost
PKI Server: Server 1
```

**On Server 2 (Alternative 2)**

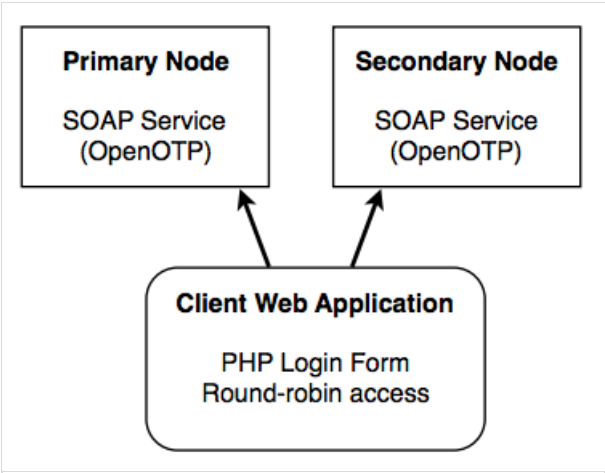
```
LDAP Servers: LDAP 2, LDAP 1
SQL Servers: SQL 2, SQL 1
Session Manager: Localhost
PKI Server: Server 1
```

## 7. Client Configuration

Your client applications using WebADM services or RADIUS can now use both cluster nodes, either at the same time with a round-robin policy for load-balancing, or in failover mode.

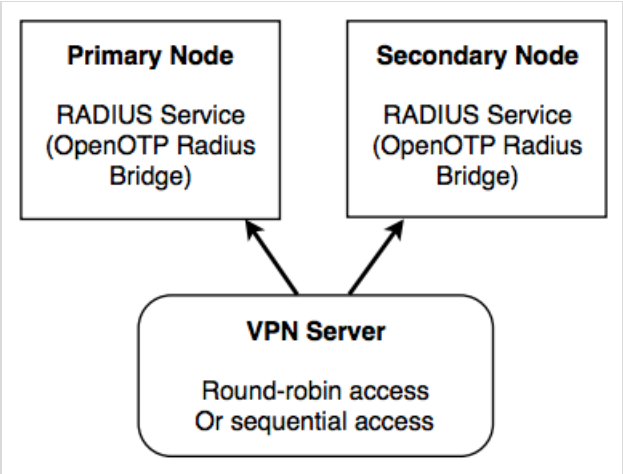
### 7.1 Web Application Using SOAP

Your web application can make SOAP calls to any web service node. With the shared session manager, client requests can come to any of the nodes, even if they are part of the same sequential OpenOTP authentication session.



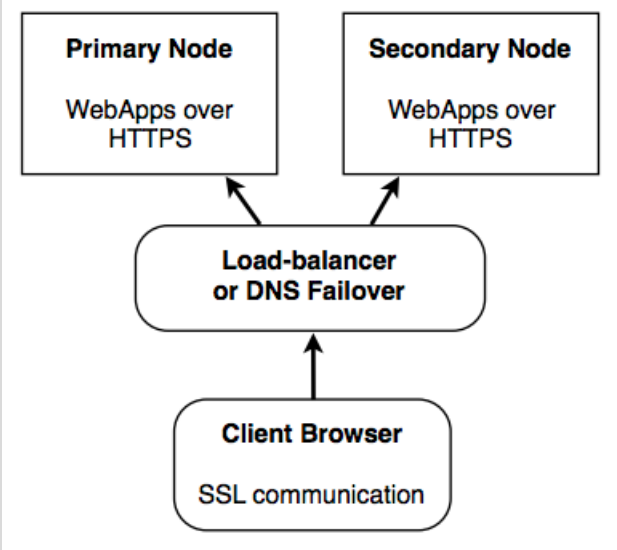

## 7.2 VPN Server Using RADIUS

The VPN client can send RADIUS requests to any of the cluster nodes. With the shared session manager, even sequential RADIUS operations such as Challenge-Responses can come to any of the nodes.




## 7.3 End-user WebApps

The WebApps can be deployed on cluster nodes like the web services and RADIUS services. The shared session manager will ensure that user sessions are opened and synchronized on any of the cluster nodes and client accesses can even come randomly to any of the cluster nodes.




## 8. Dedicated Node Roles

WebADM is composed of several components which can be assigned to a specific node in your cluster. You can disable a component (and also a role) on a node by editing the `/opt/webadm/conf/` webadm.conf file.

By default, a node has all the roles enabled:

```
enable_admin Yes
enable_manager Yes
enable_webapps Yes
enable_websrvs Yes
```

- The Admin Portal: It is preferred to have an internal node dedicated to server administration and to disable the Admin Portal on the front-end nodes especially when the HTTP services for WebApps are exposed on the Internet. If the Admin node uses the common session manager, it will be able to inform all the other nodes of an LDAP configuration change immediately (ex. OpenOTP setting update).
- The WebApps: They can be deployed on the internal network or on the public network of the company (i.e. The DMZ) to be used by the users from the Internet.
- The web services: It is preferred not to allow public access to the SOAP services and RADIUS services. You should enable connections from the local client applications only. And you should allow remote client accesses only through secure connectivity networks such as IPSec transport.

The RCDevs SMSHub web service can be deployed on one node and serve the role of internal SMS gateway when used with multiple OpenOTP service nodes.