

Server preparation - Linux

[installation](#), [java](#), [tomcat](#), [quickstart](#), [apache httpd](#)

This tutorial shows how to prepare the server for test or production usage of CzechIdM. If you are looking for much quicker way of how to start the CzechIdM, use the demo setup described here

[Getting Started](#)

Basic system setup

- 1 server (can be virtualized) for all: backend, frontend and database.
- OS Linux with EPEL repository enabled - CENTOS, basic network enabled installation
 - It is possible to use Debian but you have to adjust the installation guide a little. We tested CzechIdM installation on Stretch.
- PostgreSQL - installed from a new repository
- Java - distribution repository (OpenJDK 1.8)
- Apache Tomcat - manually installed into /opt/tomcat
- Services start via systemd in OS
- Services run under dedicated user (non-privileged one)

Instalation and software configuration

Prerequisites - Basic installation of CentOS 7

```
# EPEL installation
yum clean all
yum install -y epel-release
yum update -y
# other recommended packages installation
yum install -y net-tools nano wget mc vim-enhanced screen sysstat bzip2
ssmtp bash-completion lsof haveged nmap zip unzip psmisc telnet
policycoreutils-python
# enable haveged after OS start
systemctl start haveged.service
systemctl enable haveged.service
# remove unnecessary software
yum remove -y postfix
systemctl stop avahi-daemon.socket avahi-daemon.service
systemctl disable avahi-daemon.socket avahi-daemon.service
yum remove -y avahi-autoipd avahi
# set the hostname
hostnamectl set-hostname FQDN_server_name
hostnamectl status
# check the network configuration, be sure it is static
(/etc/sysconfig/network-scripts/)
# reboot the server
```

When installing on Debian, install these packages:

```
screen dnsutils sysstat lsof haveged nmap tcpdump traceroute tcptraceroute  
curl iptables-persistent
```

PostgreSQL



If you are install CzechIdM on Sql server, please follow [this tutorial](#).

CentOS7 default repository version of PostgreSQL is 9.2. In our tutorial, we will install newer version 9.6. Moreover, we install database data into /data not /var/lib which is the default option.

Database server installation - CentOS7

- Software installation (versions can vary):

```
yum install -y  
https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-  
centos96-9.6-3.noarch.rpm  
yum install -y postgresql96-server postgresql96-contrib pgstat2_96 pg_top96  
postgresql96-libs
```

- create new system directory:

```
mkdir -p /data/pgsql/9.6/data/  
chown -R postgres:postgres /data/pgsql/  
chmod 700 /data/pgsql
```

- Copy of the configuration file for systemd, in which we will make change of directory for data:

```
cp /usr/lib/systemd/system/postgresql-9.6.service /etc/systemd/system/
```

In the file /etc/systemd/system/postgresql-9.6.service change the directory for data as follows:

```
# Location of database directory  
Environment=PGDATA=/data/pgsql/9.6/data/
```

- In the file ~postgres/.bash_profile change the variable PGDATA value to:

```
PGDATA=/data/pgsql/9.6/data
```

- Reload changes:

```
systemctl daemon-reload
```

- Initialize database:

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

- Enable and start database:

```
systemctl start postgresql-9.6.service  
systemctl enable postgresql-9.6.service
```

- Check that the database is running:

```
[root@tomcat1 system]# systemctl status postgresql-9.6.service -l  
● postgresql-9.6.service - PostgreSQL 9.6 database server  
   Loaded: loaded (/etc/systemd/system/postgresql-9.6.service; enabled;  
   vendor preset: disabled)  
   Active: active (running) since Pá 2016-11-18 23:50:06 CET; 2min 57s ago  
   Main PID: 2626 (postmaster)  
   CGroup: /system.slice/postgresql-9.6.service  
           └─2626 /usr/pgsql-9.6/bin/postmaster -D /data/pgsql/9.6/data/  
             └─2628 postgres: logger process  
               └─2630 postgres: checkpoint process  
                 └─2631 postgres: writer process  
                   └─2632 postgres: wal writer process  
                     └─2633 postgres: autovacuum launcher process  
                       └─2634 postgres: stats collector process  
  
lis 18 23:50:06 tomcat1.localdomain systemd[1]: Starting PostgreSQL 9.6  
database server...  
lis 18 23:50:06 tomcat1.localdomain postmaster[2626]: < 2016-11-18  
23:50:06.608 CET > LOG:  redirecting log output to logging collector process  
lis 18 23:50:06 tomcat1.localdomain postmaster[2626]: < 2016-11-18  
23:50:06.608 CET > HINT:  Future log output will appear in directory  
"pg_log".  
lis 18 23:50:06 tomcat1.localdomain systemd[1]: Started PostgreSQL 9.6  
database server.
```

Database server installation - Debian Stretch

Install the database from OS packages:

```
apt-get install postgresql-9.6
```

We will move the database - create directory structure:

```
mkdir -p /data/pgsql/9.6/data/  
chown -R postgres:postgres /data/pgsql/  
chmod -R 700 /data/pgsql
```

Create the file `.bash_profile` in postgres user's home (default `/var/lib/postgresql`) with following contents:

```
PGDATA=/data/pgsql/9.6/data
```

Stop the database:

```
systemctl stop postgresql
```

Move database directory (run this as root):

```
mv /var/lib/postgresql/9.6/main/* /data/pgsql/9.6/data/
```

In the PostgreSQL configuration file `/etc/postgresql/9.6/main/postgresql.conf` set the `data_directory` property to:

```
data_directory = '/data/pgsql/9.6/data'
```

Enable and start the database:

```
systemctl start postgresql  
systemctl enable postgresql
```

DB server configuration

First of all, enable the password authentication.

In the file `/data/pgsql/9.6/data/pg_hba.conf` find lines:

host	all	all	127.0.0.1/32	ident
host	all	all	:::1/128	ident

and change the value at the end of each line into md5 like this:

host	all	all	127.0.0.1/32	md5
host	all	all	:::1/128	md5

Now we can do DB sizing. We presume the system has 3GB dedicated for the db. We can also log the queries logging (those over 200ms). **For particular sizing, use a [calculator](#).** In a file `/data/pgsql/9.6/data/postgresql.conf` edit (add those if not exist) lines:

```
max_connections = 100          # (change requires restart)  
  
shared_buffers = 768MB        # min 128kB  
effective_cache_size = 2304MB  
work_mem = 7864kB  
maintenance_work_mem = 192MB  
  
min_wal_size = 1GB  
max_wal_size = 2GB  
checkpoint_completion_target = 0.7
```

```
wal_buffers = 16MB

default_statistics_target = 100

log_min_duration_statement = 200
```

Restart DB: `systemctl restart postgresql-9.6.service`

For Debian installation, edit those configuration files instead:

```
/etc/postgresql/9.6/main/pg_hba.conf
/etc/postgresql/9.6/main/postgresql.conf
```



If you install the database to a different server than the CzechIdM application itself (Tomcat etc.), don't forget to configure PostgreSQL to allow remote SSL connection from that server.

Tomcat

Installation - CentOS7:

```
yum install -y tomcat
```

Installation - Debian:

```
apt install -y tomcat8
```

Start Tomcat automatically after system startup - CentOS

- Make some adjustments to systemd unit.

```
systemctl edit tomcat.service
```

Or if you want use different editor than nano(vim) use this commands:

```
export SYSTEMD_EDITOR="/bin/vim"
sudo -E systemctl edit tomcat.service
```

- Add these lines and save the file:

```
[Service]
SyslogFacility=local3
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -
Djava.security.egd=file:/dev/./urandom -
Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true -
```

```
Djavax.servlet.request.encoding=UTF-8'
```

- Values of Xms and Xmx are closely dependent on server sizing. If you have enough memory it is strongly advised to use Xmx 6128M or more.
- Tomcat will be started under user tomcat:tomcat.
- After every systemd configuration change it is necessary to reload:

```
systemctl daemon-reload
```

- Test start:

```
systemctl start tomcat
```

- Check that Tomcat runs with desirable parameters:

```
[root@tomcat1 logs]# ps -u tomcat -fwww
UID          PID  PPID  C  STIME TTY          TIME CMD
tomcat    14221      1   0   10:17 ?           00:00:03 /usr/lib/jvm/jre/bin/java -
Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -
Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true -
Djavax.servlet.request.encoding=UTF-8 -Xms512M -Xmx1024M -server -
XX:+UseParallelGC -classpath
/usr/share/tomcat/bin/bootstrap.jar:/usr/share/tomcat/bin/tomcat-
juli.jar:/usr/share/java/commons-daemon.jar -
Dcatalina.base=/usr/share/tomcat -Dcatalina.home=/usr/share/tomcat -
Djava.endorsed.dirs= -Djava.io.tmpdir=/var/cache/tomcat/temp -
Djava.util.logging.config.file=/usr/share/tomcat/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
org.apache.catalina.startup.Bootstrap start
```

- Stop Apache Tomcat:

```
systemctl stop tomcat
```

- Enable tomcat start after OS start:

```
systemctl enable tomcat
```

Start Tomcat automatically after system startup - Debian

- In file /etc/default/tomcat8 set the basic JVM parameters. If they are there already, change them.

tomcat8

```
CATALINA_OPTS="-Xms512M -Xmx1024M -server -XX:+UseParallelGC"
JAVA_OPTS="-Djava.awt.headless=true -
Djava.security.egd=file:/dev/./urandom -
Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true -
Djavax.servlet.request.encoding=UTF-8"
```

- Values of Xms and Xmx are closely dependent on server sizing. If you have enough memory it is strongly advised to use Xmx 6128M or more.
- Tomcat will be started under user tomcat8:tomcat8.
- Test start:

```
systemctl start tomcat8
```

- Check that Tomcat runs with desirable parameters:

```
[root@tomcat1 logs]# ps -u tomcat8 -fwww
UID          PID  PPID  C  STIME TTY          TIME CMD
tomcat8      742    1   0  13:20 ?           00:00:03 /usr/lib/jvm/java-8-openjdk-
amd64/bin/java -
Djava.util.logging.config.file=/var/lib/tomcat8/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -
Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true -
Djavax.servlet.request.encoding=UTF-8 -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Xms512M -
Xmx1024M -server -XX:+UseParallelGC -classpath
/usr/share/tomcat8/bin/bootstrap.jar:/usr/share/tomcat8/bin/tomcat-juli.jar
-Dcatalina.base=/var/lib/tomcat8 -Dcatalina.home=/usr/share/tomcat8 -
Djava.io.tmpdir=/tmp/tomcat8-tomcat8-tmp
org.apache.catalina.startup.Bootstrap start
```

- Stop Apache Tomcat:

```
systemctl stop tomcat8
```

- Enable tomcat start after OS start:

```
systemctl enable tomcat8
```

Apache Tomcat configuration recommended for production usage

It is advised to follow these steps for production usage:

- In file /etc/tomcat/server.xml (/etc/tomcat8/server.xml on debian)

- Turn off the shutdown port:
 - Set value -1 from 8005 to the Server port tag, thus you deactivate it:

```
<Server port="-1" shutdown="SHUTDOWN">
```

- In same file do this:

- Make Tomcat listen only on localhost:
 - Add the address="127.0.0.1" property to configuration of 8009 and 8080 ports.
 - In Debian you need to uncomment AJP connector on port 8009.
- Change logging into localhost_access_log.

- Find these lines and comment them.

```
<!--  
<Valve className="org.apache.catalina.valves.AccessLogValve"  
directory="logs"  
        prefix="localhost_access_log." suffix=".txt"  
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />  
-->
```

And add these lines:

```
<Valve className="org.apache.catalina.valves.AccessLogValve"  
directory="logs"  
        prefix="localhost_access_log." suffix="log"  
        pattern="%h %l %u %t &quot;%r&quot; %s %b"  
        rotatable="false" />
```

- In the file /etc/tomcat/web.xml(/etc/tomcat8/web.xml on debian)

- Do not show application server version:
 - Set showServerInfo to false (default is true):

```
<servlet>  
    <servlet-name>default</servlet-name>  
    <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-  
class>  
    <init-param>  
        <param-name>debug</param-name>  
        <param-value>0</param-value>  
    </init-param>  
    <init-param>  
        <param-name>listings</param-name>  
        <param-value>>false</param-value>  
    </init-param>  
    <init-param>  
        <param-name>showServerInfo</param-name>  
        <param-value>>false</param-value>  
    </init-param>  
    <load-on-startup>1</load-on-startup>  
</servlet>
```

We need to tell Tomcat where idm.war will be. Create context file
/etc/tomcat/Catalina/localhost/idm.xml(
/etc/tomcat8/Catalina/localhost/idm.xml on debian) with these lines:

```
<Context  
    docBase="/opt/czechidm/app/idm.war"  
    path=""  
>
```

Tomcat logging configuration

- in file /etc/tomcat/logging.properties(/etc/tomcat8/logging.properties on debian)

- Change logging properties
 - Add/change lines(1catalina, 2localhost, 3manager, 4host-manager) into this(leave the other lines as they are):

```
1catalina.org.apache.juli.FileHandler.level = ALL
1catalina.org.apache.juli.FileHandler.prefix = tomcat.
1catalina.org.apache.juli.FileHandler.rotatable = false
1catalina.org.apache.juli.FileHandler.suffix = log

2localhost.org.apache.juli.FileHandler.rotatable = false
2localhost.org.apache.juli.FileHandler.suffix = log

3manager.org.apache.juli.FileHandler.rotatable = false
3manager.org.apache.juli.FileHandler.suffix = log

4host-manager.org.apache.juli.FileHandler.rotatable = false
4host-manager.org.apache.juli.FileHandler.suffix = log
```

On Debian make these extra changes:

```
handlers = 1catalina.org.apache.juli.AsyncFileHandler,
2localhost.org.apache.juli.AsyncFileHandler,
3manager.org.apache.juli.FileHandler, 4host-
manager.org.apache.juli.FileHandler
#, java.util.logging.ConsoleHandler

.handlers = 1catalina.org.apache.juli.FileHandler
#, java.util.logging.ConsoleHandler

#####
# Handler specific properties.
#####

3manager.org.apache.juli.FileHandler.level = FINE
3manager.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
3manager.org.apache.juli.FileHandler.prefix = manager.

4host-manager.org.apache.juli.FileHandler.level = FINE
4host-manager.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
4host-manager.org.apache.juli.FileHandler.prefix = host-manager.

#java.util.logging.ConsoleHandler.level = FINE
#java.util.logging.ConsoleHandler.formatter =
org.apache.juli.OneLineFormatter

#####
```

```
# Facility specific properties.
#####

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].handlers = 3manager.org.apache.juli.FileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].handlers = 4host-manager.org.apache.juli.FileHandler
```

On CentOS for redirect logging from /var/log/messages. Add this rule into /etc/rsyslog.d/tomcat.conf file.

```
### tomcat log
$template TomcatForm,"%msg%\n"
if ($syslogfacility-text == 'local3' and $syslogtag contains "server")
then{
    action(type="omfile" file="/var/log/tomcat/catalina.out"
FileCreateMode="0644" fileOwner="tomcat" fileGroup="tomcat"
template="TomcatForm" )
    & stop
}
```

Then restart rsyslog

```
systemctl restart rsyslog
```

Rotating Tomcat logs

Tomcat logger appends to the logfile at /var/log/tomcat/. Tomcat also sets up logrotate at /etc/logrotate.d/tomcat. Change logrotate file into following and adjust log retention (the COUNT) as necessary - for production deployments we recommend at least 90 days.

tomcat

```
/var/log/tomcat/tomcat.log
/var/log/tomcat/manager.log
/var/log/tomcat/localhost_access_log.log
/var/log/tomcat/localhost.log
/var/log/tomcat/host-manager.log{
    rotate COUNT
    daily
    dateext
    copytruncate
    missingok
    notifempty
```

```
    compress
    create 0644 tomcat tomcat
}
/var/log/tomcat/catalina.out
{
    rotate COUNT
    daily
    dateext
    copytruncate
    missingok
    notifempty
    compress
    create 0644 tomcat tomcat
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2>
/dev/null || true
    endscript
}
```

On **Debian** logs are in `/var/log/tomcat8/` and logrotate config is in `/etc/logrotate.d/tomcat8`.

tomcat8

```
/var/log/tomcat8/tomcat.log
/var/log/tomcat8/manager.log
/var/log/tomcat8/localhost_access_log.log
/var/log/tomcat8/localhost.log
/var/log/tomcat8/host-manager.log
/var/log/tomcat8/catalina.out {
    rotate COUNT
    daily
    dateext
    copytruncate
    missingok
    notifempty
    compress
    create 0644 tomcat8 tomcat8
}
```

It is possible that, on some distros, SELinux will deny access to the logfile for logrotate because `logrotate_t` is only allowed in the `/var/log` and subfolders. The logrotate will error to the `/var/log/messages` line similar to `Sep 3 03:48:01 server.tld logrotate: ALERT exited abnormally with [1]`.

If this happens, set the permissive mode for logrotate:

```
semanage permissive -a logrotate_t
```

Evaluate impact of SELinux adjustments **before** you implement them. Proper mitigation heavily depends on habits and security policies of your organization.



There are some possibilities:

- Set permissive mode for logrotate as above.
- Set permissive mode for whole SELinux. (This will drop the SELinux's protective function.)
- Adjust particular SELinux labels. Example ([here](#)).

Please note that the log does not rotate during the first day, but after the second day.

Optional - Management Interface for Tomcat

If you installed two additional applications for tomcat management follow this part to complete tomcat configuration.

These applications are available at:

- <http://localhost:8080/manager>
- <http://localhost:8080/host-manager>

If you want to use them, it is necessary to do following steps.

First of all, create a database user that you will use for the access to those applications. If you plan to connect to the applications remotely (not only from localhost) you have to also allow communication from your IP.

Create user like this:

Create the a new user in the file `/etc/tomcat/tomcat-users.xml` (on Debian `/etc/tomcat8/tomcat-users.xml`) and assign him roles "manager-gui" and "admin-gui". The documentation of available roles as well as overall configuration of the application is a part of application installation available at http://localhost:8080/docs/manager-howto.html#Configuring_Manager_Application_Access

The file `/etc/tomcat/tomcat-users.xml` (on Debian `/etc/tomcat8/tomcat-users.xml`) looks like this:

`tomcat-users.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-
users.xsd"
```

```

        version="1.0">
<role rolename="manager-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<role rolename="admin-gui"/>
<user username="admin" password="*****store it somewhere safe*****"
roles="manager-gui,manager-status,admin-gui"/>
</tomcat-users>

```

If you plan to connect to the applications remotely (not only from localhost) you have to also allow communication from your IP. If you see 403 Access Denied it might be you did not do this setting.

Add your IP address into application configuration files. In files
 /var/lib/tomcat/webapps/manager/META-INF/context.xml and
 /var/lib/tomcat/webapps/host-manager/META-INF/context.xml(on Debian
 /var/lib/tomcat8/webapps/...) add net mask for your IP (both files should have the same content):

In my case, I want to access to Tomcat management from network 192.168.0.0/24:

context.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<Context antiResourceLocking="false" privileged="true" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|192\.168\.\d+\.\d+" />
</Context>

```

Again, restart the tomcat:

```
systemctl restart tomcat
```

Apache httpd as a reverse proxy

It is possible to open Apache Tomcat to the network directly, but little inconvenient. You want the users to access the CzechIdM on user-friendly ports 80/tcp or 443/tcp, which is not easy to setup in Tomcat itself running under nonprivileged user. So we use Apache httpd as a reverse proxy. Apache httpd will allow access to data via https on port 443/tcp and http on port 80/tcp. Communication via http protocol will be enabled, but we will redirect all communication to https. Communication between Apache httpd and Tomcat will take place on local machine via AJP protocol. In httpd, there will be mod_security installed (optional but recommended), which serves as an application firewall.

The configuration example is written for the server which allows access to its services under the name "demo.czechidm.com".

HTTPd installation and configuration

Install httpd and mod_security:

```
yum install -y httpd httpd-tools mod_ssl mod_security mod_security_crs
```

On Debian install those packages and allow modules:

```
apt-get install apache2 libapache2-mod-security2 modsecurity-crs
a2enmod ssl
a2enmod proxy
a2enmod proxy_ajp
a2enmod proxy_http
a2enmod security2
a2enmod rewrite
a2enmod headers
```

HTTPd basic configuration:

Change MPM to worker (lower system requirements) - in the file
/etc/httpd/conf.modules.d/00-mpm.conf comment the lines with mod_mpm_prefork.so and
uncomment mod_mpm_worker.so:

```
# Select the MPM module which should be used by uncommenting exactly
# one of the following LoadModule lines:

# prefork MPM: Implements a non-threaded, pre-forking web server
# See: http://httpd.apache.org/docs/2.4/mod/prefork.html
#LoadModule mpm_prefork_module modules/mod_mpm_prefork.so

# worker MPM: Multi-Processing Module implementing a hybrid
# multi-threaded multi-process web server
# See: http://httpd.apache.org/docs/2.4/mod/worker.html
#
LoadModule mpm_worker_module modules/mod_mpm_worker.so

# event MPM: A variant of the worker MPM with the goal of consuming
# threads only for connections with active processing
# See: http://httpd.apache.org/docs/2.4/mod/event.html
#
#LoadModule mpm_event_module modules/mod_mpm_event.so
```

Disable "welcome" page:

```
cd /etc/httpd/conf.d
mv welcome.conf welcome.conf-DISABLED
touch welcome.conf
```

Virtualhost configuration to forward the communication from port 80 to 443. Add following section

and change string 'server' to the real servername in the file `/etc/httpd/conf.d/vhost-redirect.conf` (or `/etc/apache2/sites-available/vhost-redirect.conf` for Debian):

```
<VirtualHost _default_:80>
    DocumentRoot /var/www/html
    Redirect permanent / https://server
</VirtualHost>
```

Set the proxy in the virtualhost for https (443/tcp) - at the end of the file `/etc/httpd/conf.d/ssl.conf` (or `/etc/apache2/sites-available/ssl.conf` for Debian) add following before ending "tag" VirtualHost:

```
ProxyRequests      off
ProxyPreserveHost  on
ProxyAddHeaders    on
ProxyPass / ajp://127.0.0.1:8009/
ProxyPassReverse / ajp://127.0.0.1:8009/
```

In IE 11, CzechIdM has problems with missing icons. Icons are created by special fonts and those fonts are handled badly in the IE. It is necessary to set Cache-Control HTTP header. We need to set it only for font files:

```
# workaround for bad font handling in IE 11
<LocationMatch "/idm/.*(\\.ttf|\\.woff2|\\.eot)$">
    Header set Cache-Control "no-cache, public, must-revalidate, proxy-revalidate"
</LocationMatch>
```

Identity manager CzechIdM will be available on address <https://server/idm/> It is possible to forward from `/` to `/idm/`, so that the user does not need to type the whole URL. To do so, add following lines to the virtualhost config file (ssl.conf):

```
RewriteEngine On
RewriteRule "^/$" "/idm/" [R]
```

In the file `ssl.conf` we also have to disable SSLv3. Edit the line with SSLProtocol directive:

```
SSLProtocol all -SSLv2 -SSLv3
```

On Debian, create symlinks to sites-enabled:

```
cd /etc/apache2/sites-enabled
ln -s ../sites-available/vhost-redirect.conf 01vhost-redirect.conf
ln -s ../sites-available/ssl.conf 02ssl.conf
```

Syntax check before httpd restart:

```
httpd -t -D DUMP_VHOST
```

httpd restart and reload configuration changes:

```
systemctl restart httpd
```

Enable httpd after OS start:

```
systemctl enable httpd.service
```

mod_security configuration

Mod_security files locations (on CentOS7):

- Audit log: /var/log/httpd/modsec_audit.log
- Directory with activated rules: /etc/httpd/modsecurity.d/activated_rules/
- basic configuration file for mod_security:
/etc/httpd/modsecurity.d/modsecurity_crs_10_config.conf
- The file for chosen rules deactivation: /etc/httpd/conf.d/ssl.conf

The default set of rules is relatively strict. CzechIdM cannot run with the default configuration of mod_security.

Each rule is identified by a unique ID. If you want to deactivate the whole rule, it is advised to write the rule ID into ssl.conf like this:

```
<IfModule mod_security2.c>  
    SecRuleRemoveById RULE_ID  
</IfModule>
```

Disabling mod_security rules

In the file /etc/httpd/conf.d/ssl.conf (or /etc/apache2/sites-available/ssl.conf for Debian) deactivate following rules and set their logging:

```
<IfModule mod_security2.c>  
    SecRuleRemoveById 981173  
    SecRuleRemoveById 960015  
    SecRuleRemoveById 950109  
  
    # Allow Czech signs  
    SecRuleRemoveById 981318  
    SecRuleRemoveById 981242  
    SecRuleRemoveById 960024  
    SecRuleRemoveById 981245  
  
    # Too restrictive for login format  
    SecRuleRemoveById 960035  
  
    # Needed by Websockets  
    <Location "/idm/api/v1/websocket-info/">
```

```

        SecRuleRemoveById 970901
    </Location>

    # These break Certificate Authority module
    <Location "/idm/api/v1/crt/certificates/action/validate">
        SecRuleRemoveById 960915
        SecRuleRemoveById 200003
    </Location>

    # do not log request/response body
    SecAuditLogParts ABFHZ
</IfModule>

```

mod_security configuration - CentOS7

In the file `/etc/httpd/modsecurity.d/modsecurity\_crs\_10\_config.conf`, find the rule with `id=900012` and add support for `content\_type=application/json`, `application/hal+json` and `text/plain` on the line starting with `tx.allowed\_request\_content\_type`, then allow PUT DELETE and PATCH methods on the line with `tx.allowed\_methods`. Whole rule after the changes looks like this:

```

SecAction \
  "id:'900012', \
  phase:1, \
  t:none, \
  setvar:'tx.allowed_methods=GET HEAD POST OPTIONS PUT PATCH DELETE', \
  setvar:'tx.allowed_request_content_type=application/hal+json|application/json|text/plain|application/x-www-form-urlencoded|multipart/form-data|text/xml|application/xml|application/x-amf', \
  setvar:'tx.allowed_http_versions=HTTP/0.9 HTTP/1.0 HTTP/1.1', \
  setvar:'tx.restricted_extensions=.asa/ .asax/ .ascx/ .axd/ .backup/ .bak/ .bat/ .cdx/ .cer/ .cfg/ .cmd/ .com/ .config/ .conf/ .cs/ .csproj/ .csr/ .dat/ .db/ .dbf/ .dll/ .dos/ .htr/ .htw/ .ida/ .idc/ .idq/ .inc/ .ini/ .key/ .licx/ .lnk/ .log/ .mdb/ .old/ .pass/ .pdb/ .pol/ .printer/ .pwd/ .resources/ .resx/ .sql/ .sys/ .vb/ .vbs/ .vbproj/ .vsdisco/ .webinfo/ .xsd/ .xsx/', \
  setvar:'tx.restricted_headers=/Proxy-Connection/ /Lock-Token/ /Content-Range/ /Translate/ /via/ /if/', \
  nolog, \
  pass"

```

mod_security configuration - Debian

Enable `mod\_security` configuration:

```

cd /etc/modsecurity
cp modsecurity.conf-recommended modsecurity.conf

```

Uncomment following rules in the `/etc/modsecurity/crs/crs-setup.conf` and change them

accordingly (add allowed content types and allowed HTTP methods):

```
SecAction \  
  "id:900200,\  
    phase:1,\  
    nolog,\  
    pass,\  
    t:none,\  
    setvar:'tx.allowed_methods=GET HEAD POST OPTIONS PUT PATCH DELETE' "  
  
SecAction \  
  "id:900220,\  
    phase:1,\  
    nolog,\  
    pass,\  
    t:none,\  
    setvar:'tx.allowed_request_content_type=application/x-www-form-  
urlencoded|multipart/form-data|text/xml|application/xml|application/x-  
amf|application/json|text/plain|application/hal+json' "
```

mod_deflate configuration

It is advised to set up gzip so the users get minimum of data from the frontend server. In the file `/etc/httpd/conf.d/ssl.conf` we add following configuration and restart the server:

```
<IfModule mod_deflate.c>  
  # Compress HTML, CSS, JavaScript, Text, XML and fonts  
  AddOutputFilterByType DEFLATE application/javascript  
  AddOutputFilterByType DEFLATE application/rss+xml  
  AddOutputFilterByType DEFLATE application/vnd.ms-fontobject  
  AddOutputFilterByType DEFLATE application/x-font  
  AddOutputFilterByType DEFLATE application/x-font-opentype  
  AddOutputFilterByType DEFLATE application/x-font-otf  
  AddOutputFilterByType DEFLATE application/x-font-truetype  
  AddOutputFilterByType DEFLATE application/x-font-ttf  
  AddOutputFilterByType DEFLATE application/x-javascript  
  AddOutputFilterByType DEFLATE application/xhtml+xml  
  AddOutputFilterByType DEFLATE application/xml  
  AddOutputFilterByType DEFLATE font/opentype  
  AddOutputFilterByType DEFLATE font/otf  
  AddOutputFilterByType DEFLATE font/ttf  
  AddOutputFilterByType DEFLATE image/svg+xml  
  AddOutputFilterByType DEFLATE image/x-icon  
  AddOutputFilterByType DEFLATE text/css  
  AddOutputFilterByType DEFLATE text/html  
  AddOutputFilterByType DEFLATE text/javascript  
  AddOutputFilterByType DEFLATE text/plain  
  AddOutputFilterByType DEFLATE text/xml
```

```
AddOutputFilterByType DEFLATE application/json
AddOutputFilterByType DEFLATE application/hal+json

# Remove browser bugs (only needed for really old browsers)
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4\.0[678] no-gzip
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
Header append Vary User-Agent

</IfModule>
```

Workaround for slow HTTPD shutdown

In some RHEL/CentOS versions Apache HTTPD shutsdown or restarts itself very slowly. It is caused by https://bugzilla.redhat.com/show_bug.cgi?id=906321. Workaround is to edit '/usr/lib/systemd/system/httpd.service' and add the option:

```
KillMode=none
```

Then reload systemd:

```
systemctl daemon-reload
```

It is absolutely correct to create new versions of unity in /etc, that has the option:

```
cp /usr/lib/systemd/system/httpd.service /etc/systemd/system/httpd.service
vim /etc/systemd/system/httpd.service # add parametr KillMode=none
systemctl daemon-reload
```

The patch of httpd should come soon so the first option is OK too.

SSO

If you want to enable SSO to CzechIdM, additional configuration must be done with mod_auth_kerb. See [SSO installation guide](#) for more details.

nginx as reverse proxy

In case that you want to use nginx instead of Apache httpd, the configuration is as follows.

```
server {
    listen    *:443 ssl http2;
    server_name  idm.domain.tld;
    client_max_body_size 1G;
    ssl on;
```

```
ssl_certificate      /path/to/fullchain.pem;
ssl_certificate_key  /path/to/privkey.pem;
gzip on;
gzip_proxied any;
gzip_types
    text/css
    text/javascript
    text/xml
    text/plain
    application/javascript
    application/x-javascript
    application/json;

location / {
    proxy_hide_header X-Frame-Options;
    add_header X-Frame-Options SAMEORIGIN;
    proxy_pass http://localhost:8080/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto "https";
    proxy_ssl_session_reuse off;
    proxy_redirect off;

    # WebSocket support
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
```

From:
<https://wiki.czechidm.com/> - CzechIdM Identity Manager

Permanent link:
https://wiki.czechidm.com/tutorial/adm/server_preparation?rev=1560172429

Last update: **2019/06/10 13:13**

