Server preparation - Windows

installation, java, tomcat, quickstart, apache httpd

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

Basic system setup

- 1 server (can be virtualized) for everything: backend, frontend and database.
- OS Windows, ideally W2016 and newer
- PostgreSQL - installed from EnterpriseDB
- Java - OpenJDK
- Apache Tomcat - installed by Tomcat .exe installer
- Services start via system services (services.msc)

Instalation and software configuration

Prerequisities - Basic installation of Windows Server 2016.

- Install the **Telnet Client** system feature through **Programs and Features**. This is optional but greatly helps with debugging network problems.
- Install Firefox. Also optional, but greatly helps with debugging IdM webapp problems.
- Install Git Bash. This will be essential when configuring IdM and checking its logs.
    - Use **checkout-windows, commit unix style endings**. It does not really matter, we will not use the git command for anything.:)
    - **Do not** enable integration with windows cmd.
- Disable unnecessary windows services.
- Disable Microsoft IIS if installed.

PostgreSQL

On Windows, we use EnterpriseDB PostgreSQL distribution. Recommended version is 12.x.

- For installation, basically follow the official guide.
- Set location for binaries to `C:\PostgreSQL\12`.
- Set location for database to `C:\PostgreSQL\12\data`.
- Install all components (pgAdmin, StackBuilder, etc.).
- Leave the locale at `[Default locale]`, we will set locale explicitly when creating a database.
- After installation, check the Windows services - the `postgresql-x86_64` service should be there, configured with autostart.
    - To enter services menu, get to the Start→Run and invoke `services.msc`.

Edit the PostgreSQL configuration file `C:\PostgreSQL\12\data\postgresql.conf` to make it listen on 127.0.0.1 only. Adjust the database sizing as necessary. The following example is for 6GB

RAM. Do not hasten to overwrite your PgSQL configuration out of hand! When in doubt, use a calculator.

```
# DB Version: 12
# OS Type: windows
# DB Type: web
# Total Memory (RAM): 6 GB
# CPUs num: 4
# Connections num: 100
# Data Storage: hdd

listen_addresses = '127.0.0.1'        # what IP address(es) to listen on;
max_connections = 100           # (change requires restart)
superuser_reserved_connections = 3  # (change requires restart)
shared_buffers = 512MB          # min 128kB
work_mem = 9611kB           # min 64kB
maintenance_work_mem = 384MB

wal_buffers = 16MB
max_wal_size = 4GB
min_wal_size = 1GB
checkpoint_completion_target = 0.7  # checkpoint target duration, 0.0 - 1.0

effective_cache_size = 4608MB
default_statistics_target = 100     # range 1-10000
random_page_cost = 4
effective_cache_size = 4608MB
max_worker_processes = 4
max_parallel_workers_per_gather = 2
max_parallel_workers = 4
max_parallel_maintenance_workers = 2

track_io_timing = on
log_autovacuum_min_duration = 0
```

Configure the authentication in the `C:\PostgreSQL\12\data\pg_hba.conf` to accept passwords. The basic configuration file should then look like this:

```
# TYPE  DATABASE        USER            CIDR-ADDRESS            METHOD

# IPv4 local & remote connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
```

If you install the database onto a server distinct from the one on which the CzechIdM application itself (Tomcat etc.) is deployed, don't forget to configure PostgreSQL to allow remote SSL connection from that server.

Java

Install the openjdk (preferred version is 11.0.2, for CzechIdm 13.1.0+ Java 21 is needed). You can download it from here. Be sure to download the **JDK**, and not only **JRE**.

OpenJDK Installation

Crete directory `C:\Program Files\Java\` and extract there downloaded zip. Then set path and JAVA HOME:

- Open the **sysdm.cpl** (Win+r ant type sysdm.cpl) dialogue and navigate to `> Advanced > Environment Variables`
- Add this line to PATH variable.

```
%JAVA_HOME%\bin
```

- Add new variable `JAVA_HOME` with value `C:\Program Files\Java\jdk-11.0.2`
- Then run `java -version` from the windows cmd to check if it is working properly.

Tomcat

Download and install the latest 9.0 branch of Apache Tomcat from here. Use the Windows installer.

- Agree with licence agreement
- Deselect `Manager` from components and click `Next`
- Set shutdown port to "-1" and connector port to "8080" and click `Next`
- Modify the `JRE path` to make it point to installed openjde **JDK** and click `Next`
- Leave the installation path on default and click `Install`
- When it's done deselect option "start tomcat" and click `finish`.

You can also use Tomcat 9.0.45 and newer, those installers set the service account to `Local Service` themselves.</note>

After installation, run the **Monitor Tomcat** application from the Start menu (or run `Tomcat9w.exe` from the Tomcat bin directory - usually `C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin`). Configure following settings:

- `initial memory pool: 512MB` (example for about 5GB RAM).
- `maximum memory pool: 4096MB` (example for about 5GB RAM).
- Add `C:\CzechIdM\etc;C:\CzechIdM\lib;C:\CzechIdM\lib\*;` to the **beginning of the**CLASSPATH.

Configure addresses the server will listen on. Open the `C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\server.xml` configuration file in the Tomcat installation. Make these changes:

- Add `address="127.0.0.1"` to the **8080/tcp** and **8009/tcp** connectors. This will make

Tomcat listen only on localhost.
- Change port number `8005` to `-1` at the Shutdown Port setting. This will effectively turn off the shutdown port.
- In the section for `Connector protocol="HTTP/1.1"` on port 8080, add the `maxSwallowSize="-1"` property.
- Uncomment the section `Connector protocol="AJP/1.3"` for the port 8009 and adjust the address parameter: `address="127.0.0.1",secretRequired="true"` and `secret="password for ajp port"` . It should look like this:

```
<Connector protocol="AJP/1.3"
           address="127.0.0.1"
           port="8009"
           redirectPort="8443"
           secretRequired="true"
           secret="***password for ajp port***"
           />
```

Use the **services.msc** dialogue to set the Apache Tomcat `StartupType` to `Automatic (Delayed Start)`. This will make the application container start after the PostgreSQL database.

`For production use, we strongly advise to remove all Tomcat's management` applications from the container.

- Locate the `webapps` folder in the Tomcat installation and delete everything that is inside.

For roles and advanced management configuration, please see the relevant chapters in the Server Preparation - Linux tutorial.

Change Tomcat logging properties

In order to set-up log rotation we need stop logging to stdout and start logging to catalina.log .

After change of these tomcat setting and setting-up **logback-spring.xml** in create_czechidm_configuration IdM will log into catalina.out and after day rotate it into catalina.YYYY-MM-DD.log. Tomcat engine will log into tomcat.log and no totate because there are only log of tomat start. Tomcat will log logs of IdM connectors into tomcat-stdout tomcat9-stdout.YYYY-MM-DD.log - it's well knows bug with workaround to make manual log rotation.

Make these changes in file `C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\logging.properties`: Comment out console handler. We don't want tomcat to log to stdout or stderr and other unnesesary logs.

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

```
manager.org.apache.juli.AsyncFileHandler

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

###2localhost.org.apache.juli.AsyncFileHandler.level = FINE
###2localhost.org.apache.juli.AsyncFileHandler.directory =
${catalina.base}/logs
###2localhost.org.apache.juli.AsyncFileHandler.prefix = localhost.
###2localhost.org.apache.juli.AsyncFileHandler.maxDays = 90
###2localhost.org.apache.juli.AsyncFileHandler.encoding = UTF-8

###3manager.org.apache.juli.AsyncFileHandler.level = FINE
###3manager.org.apache.juli.AsyncFileHandler.directory =
${catalina.base}/logs
###3manager.org.apache.juli.AsyncFileHandler.prefix = manager.
###3manager.org.apache.juli.AsyncFileHandler.maxDays = 90
###3manager.org.apache.juli.AsyncFileHandler.encoding = UTF-8

###4host-manager.org.apache.juli.AsyncFileHandler.level = FINE
###4host-manager.org.apache.juli.AsyncFileHandler.directory =
${catalina.base}/logs
###4host-manager.org.apache.juli.AsyncFileHandler.prefix = host-manager.
###4host-manager.org.apache.juli.AsyncFileHandler.maxDays = 90
###4host-manager.org.apache.juli.AsyncFileHandler.encoding = UTF-8

###java.util.logging.ConsoleHandler.level = FINE
###java.util.logging.ConsoleHandler.formatter =
org.apache.juli.OneLineFormatter
###java.util.logging.ConsoleHandler.encoding = UTF-8

#org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
#org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers =
2localhost.org.apache.juli.AsyncFileHandler

#org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].le
vel = INFO
#org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].ha
ndlers = 3manager.org.apache.juli.AsyncFileHandler

#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.AsyncFileHandler
```

In 1catalina file handler change log level to "INFO" and prefix from "catalina" to "tomcat". Also set property rotable to "false". Tomcat write to this file only when starting or shutting down.

```
#1catalina.org.apache.juli.AsyncFileHandler.level = FINE
#1catalina.org.apache.juli.AsyncFileHandler.prefix = catalina.
1catalina.org.apache.juli.AsyncFileHandler.level = INFO
```

```
1catalina.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
1catalina.org.apache.juli.AsyncFileHandler.prefix = tomcat
1catalina.org.apache.juli.AsyncFileHandler.rotatable = false
1catalina.org.apache.juli.AsyncFileHandler.suffix = .log
```

Then in file `C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\server.xml` remove creation of access logs. Coment out org.apache.catalina.valves.AccessLogValve.

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

Apache httpd as a reverse proxy

It is possible to open Apache Tomcat to the network directly, but somewhat inconvenient. You want the users to access CzechIdM on user-friendly ports 80/tcp or 443/tcp. So we use Apache httpd as a reverse proxy and add a few security features along the way. Apache httpd will allow access to data via https on port 443/tcp and http on port 80/tcp. Communication via http protocol is enabled, but we redirect all communication to https. Communication between Apache httpd and Tomcat takes 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

First, install necessary VCredist library.

Download Apache HTTPd from the apachelounge distribution and unpack it into `C:\apache24` folder.

Fire up an elevated shell and install the Apache HTTPd service:

```
cd C:\apache24\bin
httpd.exe -k install
```

Open the **services.msc** and reconfigure "Apache2.4" service:

- To have `StartupType=Automatic (Delayed start)`.
- To execute under `Local Service` user. (On the "Log On" card, set user to `Local Service`, delete contents of password fields and click Apply.)

Configure the HTTPd in its core config file `C:\Apache24\conf\httpd.conf`.

httpd.conf

```
#uncomment these modules
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule deflate_module modules/mod_deflate.so
LoadModule filter_module modules/mod_filter.so
LoadModule http2_module modules/mod_http2.so
LoadModule headers_module modules/mod_headers.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule unique_id_module modules/mod_unique_id.so

#add modsecurity module:
    LoadModule security2_module modules/mod_security2.so

#change ServerName and Server Admin
ServerAdmin root@demo.czechidm.com
ServerName demo.czechidm.com

#uncomment include vhosts a ssl configuration
Include conf/extra/httpd-vhosts.conf
Include conf/extra/httpd-ssl.conf

# Include modsec configuration if module is loaded
<IfModule mod_security2.c>
    Include conf/extra/modsec.conf
</IfModule>
```

Configure the HTTP→HTTPS redirect in the `C:\Apache24\conf\extra\httpd-vhosts.conf`. Replace `demo.czechidm.com` with the name of your server. Comment other template vhost which you don't need:

httpd-vhosts.conf

```
# Virtual Hosts
#
# Required modules: mod_log_config

<VirtualHost *:80>
    ServerName demo.czechidm.com
    ErrorLog "logs/demo.czechidm.com-error.log"
    CustomLog "logs/demo.czechidm.com-access.log" common

# this is for stable deployment
    Redirect permanent / https://demo.czechidm.com
```

```
# this one is for debugging before going live
#    Redirect / https://demo.czechidm.com
</VirtualHost>
```

Configure the HTTPS virtual host in the `C:\Apache24\conf\extra\httpd-ssl.conf` file. Change `demo.czechidm.com` to the name of your server.

In some cases older clients (i.e. IE10 and older, Java6, etc.) will not be able to communicate with IdM due to the SSL settings. If this is your case, you may need to slacken the cipher settings a bit.

httpd-ssl.conf

```
Listen 443

SSLCipherSuite
ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:!LOW:!RC4:!3DES+SHA:!IDEA
SSLProxyCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES
SSLHonorCipherOrder on
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLProxyProtocol all -SSLv2 -SSLv3
SSLPassPhraseDialog  builtin
SSLSessionCache        "shmcb:c:/Apache24/logs/ssl_scache(512000)"
SSLSessionCacheTimeout  300

<VirtualHost _default_:443>
ServerName demo.czechidm.com
ServerAdmin root@demo.czechidm.com
ErrorLog "c:/Apache24/logs/demo.czechidm.com_ssl-error.log"
TransferLog "c:/Apache24/logs/demo.czechidm.com_ssl-access.log"
CustomLog "c:/Apache24/logs/demo.czechidm.com_ssl-request.log" "%t %h
%{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

SSLEngine on

SSLCertificateFile "c:/Apache24/conf/server.crt"
SSLCertificateKeyFile "c:/Apache24/conf/server.key"
#SSLCertificateChainFile "c:/Apache24/conf/server-chain.crt"

SSLVerifyClient none

<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory "c:/Apache24/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>
```

```
BrowserMatch "MSIE [2-5]" nokeepalive ssl-unclean-shutdown
downgrade-1.0 force-response-1.0

# 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>

Protocols       https/1.1
ProxyRequests     off
ProxyPreserveHost on
ProxyAddHeaders on
ProxyPass / ajp://127.0.0.1:8009/ secret=**tomcat_ajp_secret**
ProxyPassReverse / ajp://127.0.0.1:8009/ secret=**tomcat_ajp_secret**

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

<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">
                SecRuleRemoveById 960915
                SecRuleRemoveById 200003
        </Location>

    # Modsec can throw false positives on some files due to multipart
boundary check
    <Location "/idm/api/v1/attachments/upload">
        SecRuleRemoveById 960915
        SecRuleRemoveById 200003
    </Location>
```

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

<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>

</VirtualHost>
```

Supply SSL certificate and key in x509 PEM form to `c:/Apache24/conf/server.key` and `c:/Apache24/conf/server.crt` files. Apache HTTPd will not start without those files. If you need to generate some ad-hoc certificates, use for example this guide. You can easily invoke the **openssl** tool from the Git Bash prompt.

Self-signed cert and key for testing purposes can be created like this:

```
openssl req -x509 -newkey rsa:4096 -keyout server.key -out server.crt -days
365 -nodes
```

mod_security installation

Download the mod_security module v 2.x from the Apache Lounge project. Unpack the zip and perform following actions:

- Copy the `mod_security2.so` int `C:\Apache24\modules` directory.
- Copy `yajl.dll` into `C:\Apache24\bin` directory.

Create general mod_security configuration file `C:\Apache24\conf\extra\modsec.conf`:

modsec.conf

```
<IfModule mod_security2.c>
    # ModSecurity Core Rules Set configuration
    IncludeOptional conf/modsecurity_win.d/*.conf
    IncludeOptional conf/modsecurity_win.d/activated_rules/*.conf

    # Default recommended configuration
    SecRuleEngine On
    SecRequestBodyAccess On
    SecRule REQUEST_HEADERS:Content-Type "text/xml" \
"id:'200000',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProce
ssor=XML"
    SecRequestBodyLimit 13107200
    SecRequestBodyNoFilesLimit 131072
    SecRequestBodyInMemoryLimit 131072
    SecRequestBodyLimitAction Reject
    SecRule REQBODY_ERROR "!@eq 0" \
    "id:'200001', phase:2,t:none,log,deny,status:400,msg:'Failed to
parse request body.',logdata:'%{reqbody_error_msg}',severity:2"
    SecRule MULTIPART_STRICT_ERROR "!@eq 0" \
    "id:'200002',phase:2,t:none,log,deny,status:44,msg:'Multipart
request body \
    failed strict validation: \
    PE %{REQBODY_PROCESSOR_ERROR}, \
    BQ %{MULTIPART_BOUNDARY_QUOTED}, \
    BW %{MULTIPART_BOUNDARY_WHITESPACE}, \
    DB %{MULTIPART_DATA_BEFORE}, \
    DA %{MULTIPART_DATA_AFTER}, \
    HF %{MULTIPART_HEADER_FOLDING}, \
    LF %{MULTIPART_LF_LINE}, \
    SM %{MULTIPART_MISSING_SEMICOLON}, \
    IQ %{MULTIPART_INVALID_QUOTING}, \
    IP %{MULTIPART_INVALID_PART}, \
    IH %{MULTIPART_INVALID_HEADER_FOLDING}, \
    FL %{MULTIPART_FILE_LIMIT_EXCEEDED}'"

    SecRule MULTIPART_UNMATCHED_BOUNDARY "!@eq 0" \
```

```
    "id:'200003',phase:2,t:none,log,deny,status:44,msg:'Multipart
parser detected a possible unmatched boundary.'"

    SecPcreMatchLimit 1000
    SecPcreMatchLimitRecursion 1000

    SecRule TX:/^MSC_/ "!@streq 0" \
            "id:'200004',phase:2,t:none,deny,msg:'ModSecurity internal
error flagged: %{MATCHED_VAR_NAME}'"

    SecResponseBodyAccess Off
#    SecDebugLog /var/log/httpd/modsec_debug.log
#    SecDebugLogLevel 0
    SecAuditEngine RelevantOnly
    SecAuditLogRelevantStatus "^(?:5|4(?!04))"
    SecAuditLogParts ABIJDEFHZ
    SecAuditLogType Serial
    SecAuditLog logs/modsec_audit.log
    SecArgumentSeparator &
    SecCookieFormat 0
    SecTmpDir modsec_tmp
    SecDataDir modsec_lib
</IfModule>
```

Create empty directories `C:\Apache24\modsec\_tmp` and `C:\Apache24\modsec\_lib` for mod_security working data.

Mod_security will become operational but will have no filtering rules. To obtain filtering rules, please visit Mod Security project git. **Remember to obtain 2.x version of rules, not the newest 3.x version!**

Create directory `C:\Apache24\conf\modsecurity\_win.d\activated\_rules`. From downloaded zip copy all rules from `coreruleset-2.2.9.zip\coreruleset-2.2.9\base\_rules` to `C:\Apache24\conf\modsecurity\_win.d\activated\_rules`. Then downloaded zip copy rule configuration `coreruleset-2.2.9.zip\coreruleset-2.2.9\modsecurity\_crs\_10\_setup.conf.example` to `C:\Apache24\conf\modsecurity\_win.d\modsecurity\_crs\_10\_config.conf`.

For commercial deployment of CzechIdM, we have prepared a pack of mod_security rules which you need to just unpack into `C:\Apache24\conf` directory, where it creates a `modsecurity\_win.d` folder full of rules.

Mod Security rules package

(login required).

Now in file `C:\Apache24\conf\modsecurity\_win.d\modsecurity\_crs\_10\_config.conf` change rule `900012` to look 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/x-www-form-
urlencoded|multipart/form-data|text/xml|application/xml|application/x-
amf|application/json|application/hal+json|text/plain', \
  setvar:'tx.allowed_http_versions=HTTP/0.9 HTTP/1.0 HTTP/1.1 HTTP/2.0', \
  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"
```

Now you can start the Apache HTTPd using its service. If it fails to start, check the Windows EventLog for errors.