Single KeyScaler Instance Stack - Prerequisites KS 6.7.3 for CentOS 7.5/RHEL 7.9

Document Name	Single Tenant - Prerequisites KS 6.7.3 CentOS 7.5 and RHEL 7.9
Version	Published
Date Published	October 2021
Classification	External Document
Author	Frode Nilsen, Nirmal Misra and Rajesh Dubey

- 1 This Document
  - 1.1 Document Version Control
  - 1.2 Assumptions
  - 1.3 Constraints
  - 1.2 Terms and Definitions
  - 1.3 Related Documentation
- 2. Introduction
  - 2.1. Document Overview
  - 2.2 Download the installation package from the Device Authority Customer Portal
  - 2.3 Run the install script
    - 2.3.1 Unpack downloaded file
    - 2.3.2 Edit the Install Script for RHEL 7.9
    - 2.3.3 Run the install Script
  - 2.4 Running over SSL or HTTPS
    - 2.4.1 Obtaining SSL Certificate
      - 2.4.1.1 Self-signed certificate
        - 2.4.1.1.1 Create temporary directory
        - 2.4.1.1.2 Generate public and private key pair
        - 2.4.1.1.3 Generate a certificate request with the public key included
        - 2.4.1.1.4 Generate a self-signed certificate
        - 2.4.1.1.5 Package the private key and certificate together as a PKCS#12
        - 2.4.1.1.6 Importing the Self-Signed Certificate to the Java Keystore
        - 2.4.1.1.7 Configure Tomcat SSL Connector
      - 2.4.1.2 CA Signed certificate
        - 2.4.1.2.1 Configure Tomcat SSL Connector
  - 2.5 Change MySQL Password
  - 2.6 Start the Tomcat Webserver
  - 2.7 Verify access to wizard pages
- 3 Next Steps

- APPENDIX XX

1 This Document

1.1 Document Version Control

Version	Description	Date	Author
1.0	Initial Document Creation	2017	Frode Nilsen
2.0	Updated for KS 6.7.0	18th September	Nirmal Misra
2.1	Updated to Include Install Instructions for RHEL 7.9	1st August 2021	Rajesh Dubey
2.2	Updated for KS 6.7.3		

Item 1 - Document Version Control

1.2 Assumptions

No	Description

Item 2 - Assumptions

1.3 Constraints

No	Description

Item 3- Constraints

1.2 Terms and Definitions

Term	Meaning
KMS	Key Management Store
DAE	Device Authority Engine
CP	Control Panel
DDKG	Dynamic Device Key Generator

Item 4 - Terms and Definitions

1.3 Related Documentation

Doc #	Title	Comment

[1]	DAKS-6.7.0-INST-KEYSCALER.docx	KeyScaler Installation document

Item 5 – Related Documentation

2. Introduction

2.1. Document Overview

This document covers the installation prerequisites for KeyScaler's server components of the Device Authority Engine (DAE), Key Management Services (KMS) and Control Panel (CP)..

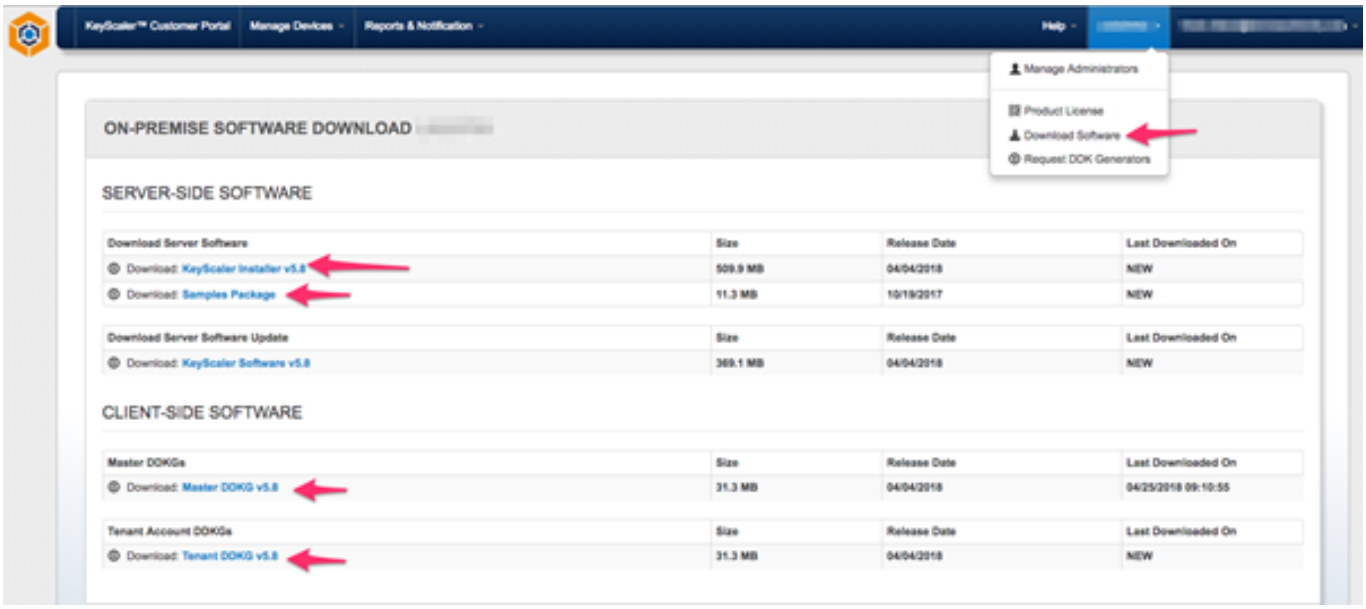
2.2 Download the installation package from the Device Authority Customer Portal

Note. This installation document assumes all files are placed in the server's home directory,

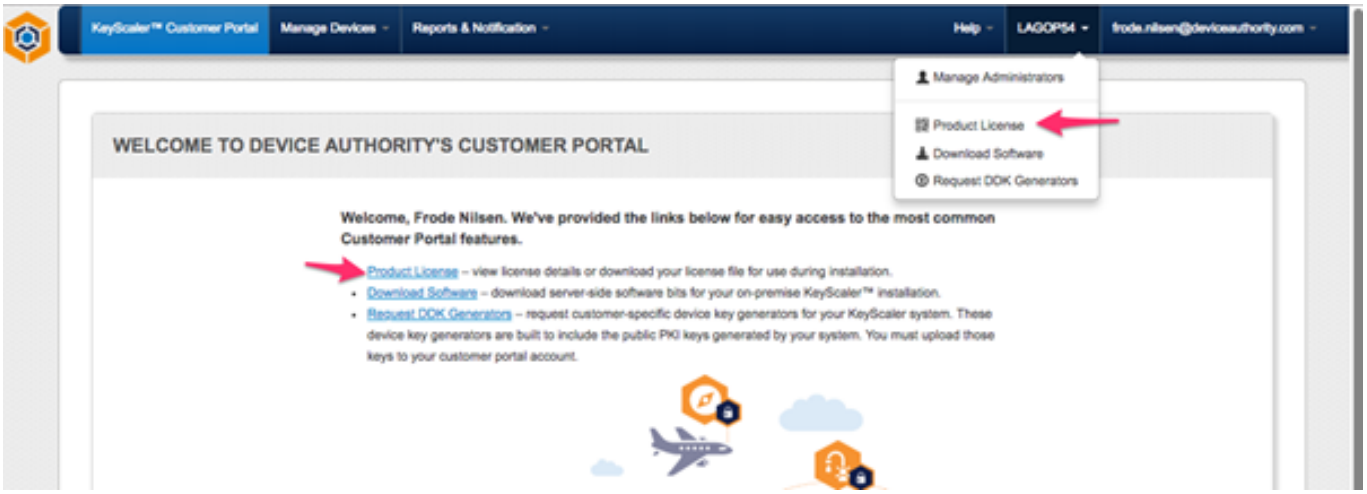
e.g. */home/ec2-user*.

Access the Device Authority Customer Portal (also called the Device Authority Download Center), navigate to the pull-down menu under your tenant name, and select Download Software. Download all the software pointed to by red arrows and transfer all of these files onto your target server to /home/ec2-user

Please note, the download links will as of September 2020, read as v6.7 and not v5.8 (below screenshot is from an old system)



Navigate to the pull-down menu under your tenant name, download the Product License and upload it to your target server directory /home/ec2-user



2.3 Run the install script

Note: if you are installing on an AWS instance and have access using ec2-user, first issue the command sudo su to become root. These instructions assume you transferred the tar.gz file to the /home/ec2-user directory.

2.3.1 Unpack downloaded file

```
[ec2-user@ip-172-31-4-186 ~]$ sudo su - root
[root@ip-172-31-4-186 ~]# cd /home/centos/
[root@ip-172-31-4-186 ec2-user]# tar -xvzf keyscaler.installer-6.5.2.tar.gz
```

Item 8 – Unpack software

If you are on **CENTOS 7.5** go ahead and continue with section 2.3.3 Run the Install Script. However if your OS is **RHEL 7.9**, Follow Section 2.3.2 first.

2.3.2 Edit the Install Script for RHEL 7.9

During install on RHEL 7.9 OS, (Note: RHEL 8.0 is not supported yet), the tomcat permissions do not get properly set, so its has to be manually set afterwards otherwise dfactor services will not start.

Edit the file install.sh by going to the bottom of the script:

Comment out the "verify_centos"

```
[root@ip-172-31-4-186 ec2-user]# cd installer
[root@ip-172-31-4-186 installer]# vi ./install.sh
# verify_centos
```

Item 8.1 – Comment out verify_centos in install.sh

Next if there are any issues with permissions when installing on RHEL 7.9, then please refer to Appendix XX to set the correct permissions up.

2.3.3 Run the install Script

```
[root@ip-172-31-4-186 ec2-user]# cd installer
[root@ip-172-31-4-186 installer]# ./install.sh
```

Item 9 – Installation script

2.4 Running over SSL or HTTPS

For basic security measures, **Tomcat must be configured to listen over SSL (HTTPs)**. All Device Authority server applications expect to send and receive traffic on a secure channel when deployed.

The Device Authority support team will refer SSL support to the institution that issues the Certificate. The SSL-related instructions below are provided as a reference only.

2.4.1 Obtaining SSL Certificate

You will need to obtain an SSL certificate. An SSL certificate is required in order for SSL to work in Tomcat. There are two ways to obtain one, and each is described below:

> **Please note:** A default password '**mypassword'** is used for the purpose of readability of this document. This password must, however, be updated in line with your company's security policy.

2.4.1.1 Self-signed certificate

> If you are using CA Signed Certificate, skip this section and proceed with 0. Please note if the KeyScaler system sits behind a load balancer in the cloud it is common practise to use self-signed certificate between the instances that reside within the VPC.

Self-signed certificates are useful in cases where you require encryption but do not need to verify the website identity. They are commonly used for testing and use on internal corporate networks (intranets). Due to the certificate not being signed by a Certification Authority (CA), users will get prompted that the site is untrusted and may have to perform several steps to "accept" the certificate before they can access the site.

The following steps illustrate the creation of a self-signed certificate + key pair converted to PKCS#12 file format (. p12). This file format allows the certificate file to be portable and reused on other Device Authority application servers running Tomcat.

2.4.1.1.1 Create temporary directory

Change to root user – create and move to a temporary directory that will be used to create files generated by the OpenSSL utility. In this example, we're using ***/var/dfactor/cert***.

```
[root@ip-172-31-4-186 installer]# mkdir /var/dfactor/cert
[root@ip-172-31-4-186 installer]# cd /var/dfactor/cert
```

Item 10 – KeyScaler Server

```
[root@ip-172-31-4-186 cert]#  yum install openssl -y
```

Item 11 - KeyScaler Server

2.4.1.1.2 Generate public and private key pair

Generate a public and a private key pair. Important Note: you'll be prompted to enter a pass phrase for the private key. You will need to remember this pass phrase for the subsequent steps.

```
[root@ip-172-31-4-186 cert]# openssl genrsa -des3 -out my_private.key 4096
Enter pass phrase for my_private.key: mypassword
Verifying - Enter pass phrase for my_private.key: mypassword
```

Item 12 - KeyScaler Server

2.4.1.1.3 Generate a certificate request with the public key included

Generate a certificate request with the public key included. You'll be prompted to enter a Common Name, make sure to use a wildcard domain name like ***.keyscaler-672-001.com** (Item 13) so that you can later use it access the other KeyScaler server applications securely using the same self-signed certificate. For example, https://tenant.keyscaler-672-001.com

> ⚠ **Please note:** The below values are sample values and should be updated to match your own company details.

```
[root@ip-172-31-4-186 cert]# openssl req -new -key my_private.key -out my_csr.csr
Enter pass phrase for my_private.key: mypassword


Country Name (2 letter code) [XX]:UK
State or Province Name (full name) []:BE
Locality Name (eg, city) [Default City]:Reading
Organization Name (eg, company) [Default Company Ltd]: DA
Organizational Unit Name (eg, section) []:Eng
Common Name (eg, your name or your server's hostname) []:*.keyscaler-672-001.com
Email Address []: your.email@yourcompany.com


A challenge password []:mypassword
An optional company name []:test
```

Item 13 - KeyScaler Server

2.4.1.1.4 Generate a self-signed certificate

Generate a self-signed certificate based on the certificate request. The example below creates a certificate good for one year.

```
[root@ip-172-31-4-186 cert]# openssl x509 -req -days 365 -in my_csr.csr -signkey my_private.key -out my_crt.crt
Signature ok
subject=/C=UK/ST=BE/L=Reading/O=DA/OU=Eng/CN=*.keyscaler-672-001.com/emailAddress=your.email@yourcompany.com
Getting Private key
Enter pass phrase for my_private.key: mypassword
```

Item 14 - KeyScaler Server

2.4.1.1.5 Package the private key and certificate together as a PKCS#12

At this point, the following you should have the following are present: **my_private.key**, **my_csr.csr**, and **my_crt.crt**. You are ready to package the private key and certificate together as a **PKCS#12** file.

Important Note: You'll be prompted to enter an Export Password for the p12 file. You will need to remember this password when configuring Tomcat's SSL Connector.

Note: The password set in below Item 15 is the password that needs to be used in the server.xml file in Item 20, when using a self-signed certificate.

```
[root@ip-172-31-4-186 cert]# openssl pkcs12 -export -inkey my_private.key -in my_crt.crt -out self_sign_certificate.p12

Enter pass phrase for my_private.key: mypassword

Enter Export Password: mypassword

Verifying - Enter Export Password: mypassword

[root@ip-172-31-4-186 cert]#
```

Item 15 - KeyScaler Server

At this stage, a **self_sign_certificate.p12** file is created with a corresponding password.

```
[root@ip-172-31-4-186 cert]# ls /var/dfactor/cert/

my_crt.crt  my_csr.csr  my_private.key  self_sign_certificate.p12
```

Item 16 - KeyScaler Server

Next, follow the instructions in the Configuring Tomcat's SSL Connector section below.

2.4.1.1.6 Importing the Self-Signed Certificate to the Java Keystore

NOTE: The instructions in this step pertain to self-signed certificates only – you may skip this step if you have configured Tomcat to use a CA signed certificate.

As root user, import your self-signed certificate to the Java keystore. **Please note** 'changeit' is the default password and *should* be updated to comply with your company's security policies for a production system. This document does not provide details on how to change this password, please consult online documentation (https://docs.oracle.com) on how to.

```
[root@ip-172-31-4-186 cert]# cd /var/dfactor/cert

[root@ip-172-31-4-186 cert]# /usr/java/default/bin/keytool -import -alias dfactor -file my_crt.crt -keystore /usr/java/latest/lib/security/cacerts -storepass changeit

Trust this certificate? [no]:  y
```

Item 17 - KeyScaler Server - Import Self-Signed Cert to KeyStore

Verify the self-signed certificate was added:

```
[root@ip-172-31-4-186 cert]# /usr/java/latest/bin/keytool -list -keystore /usr/java/default/lib/security/cacerts -storepass changeit | grep dfactor

dfactor, Mar 15, 2021, trustedCertEntry,
```

Item 18 - KeyScaler Server - Verify Self-Signed Cert added in KeyStore

The self-signed certificate is now valid for any host under the domain that is associated with the common name created in ***.keyscaler-672-001.com.**

2.4.1.1.7 Configure Tomcat SSL Connector

At this stage, you have an SSL Certificate, and need to configure Tomcat to use it when sending/receiving traffic over SSL/HTTPs. Update the **server.xml** file to specify the name and location of your **p12** file along with the keystore password you supplied when creating the **p12** file.

Edit the file **/var/www/tomcat/conf/server.xml** and add the following connector definition.

```
[root@ip-172-31-4-186 cert]# vi /var/www/tomcat/conf/server.xml
```

Item 19 - KeyScaler Server - Edit Server.xml

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
        SSLEnabled="true"
        scheme="https"
        secure="true"
        clientAuth="false"
        sslProtocol="TLS"

        maxHttpHeaderSize="8192"
        maxThreads="150"
        minSpareThreads="25"
        enableLookups="false"
        disableUploadTimeout="true"
        acceptCount="100"
        useBodyEncodingForURI="true"

        keystoreType="pkcs12"
        keystoreFile="/var/dfactor/cert/self_sign_certificate.p12"
        keystorePass="mypassword" />
```

Item 20 - KeyScaler Server – Note the password set here is must match the password set in Item 15

The important thing to specify is that **keystoreType="pkcs12"**, the keystorePass is the export password you gave when generating **pkcs12** file in the earlier section, and the **keystoreFile** is the path to the file.

2.4.1.2 CA Signed certificate

If you are using Self Signed Certificate, this section can be skipped.

If you have a CA Signed Certificate and in the format of a keystore file, it should simply be a matter of uploading the keystore file, e.g. **myKeystore** to **/var/www/tomcat/conf** and enter the relevant password in the **/var/www /tomcat/conf/server.xml** file shown below. Ensure also the keystoreType is **JKS**

2.4.1.2.1 Configure Tomcat SSL Connector

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
        SSLEnabled="true"
        scheme="https"
        secure="true"
        clientAuth="false"
        sslProtocol="TLS"

        maxHttpHeaderSize="8192"
        maxThreads="150"
        minSpareThreads="25"
        enableLookups="false"
        disableUploadTimeout="true"
        acceptCount="100"
        useBodyEncodingForURI="true"

        keystoreType="JKS"
        keystoreFile="conf/myKeystore"
        keystorePass="mypassword" />
```

Item 21 - KeyScaler Server

2.5 Change MySQL Password

IMPORTANT: If you need to change MySQL passwords once KeyScaler is installed and running, please contact customer_support@deviceauthority.com for assistance as the following instructions are only applicable prior to installing KeyScaler.

To change the default passwords used for MySQL, use the following commands, supplying your new passwords as indicated. The default root password for MySQL is admin.

REMEMBER these passwords as you'll be using them to install and administer KeyScaler.

⚠ **Please note:** A default password ‘mypassword’ for the purpose of readability of this document. These passwords must, however, be updated in line with your company’s security policy.

⚠ Please note: If you are using AWS RDS MySQL database the below steps (Item 22, Item 23) can be skipped

```
[root@ip-172-31-4-186 cert]# mysql -u root -p

Enter password: admin
mysql>  use mysql;
update user set password=PASSWORD('mypassword') where User='root';
update user set password=PASSWORD('mypassword') where User='dfactor_user';
flush privileges;
quit
```

Item 22 - KeyScaler Server – change database password

Then, verify access by logging in using both users and their new passwords.

```
[root@test ]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.5.30 MySQL Community Server (GPL)
...
mysql> quit

[root@test ]# mysql -u dfactor_user -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
...
mysql> quit
```

Item 23 - KeyScaler Server – Database Login

2.6 Start the Tomcat Webserver

As root, start the service (called dfactor):

```
[root@test ]# service dfactor start
Starting DeviceAuthority D-Factor
Using DFACTOR_HOME: /var/dfactor
Using IDP_HOME: /var/dfactor/idp
Using CATALINA_BASE:   /var/www/tomcat
Using CATALINA_HOME:   /var/www/tomcat
Using CATALINA_TMPDIR: /var/www/tomcat/temp
Using JRE_HOME:        /usr/java/latest
Using CLASSPATH:       /var/www/tomcat/bin/bootstrap.jar:/var/www/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

Item 24 – start dfactor service

You can tail the Tomcat logs to verify the test page located in ROOT/index.html under **/var/www/tomcat/webapps/** has deployed:

```
[root@test ~]# tail -f /var/www/tomcat/logs/catalina.out


2017-02-14 19:39:57,122 localhost-startStop-1 ERROR RollingFile contains an invalid element or attribute "suppressExceptions"

14-Feb-2017 19:39:58.608 INFO [localhost-startStop-1] org.apache.catalina.util.SessionIdGeneratorBase.createSecureRandom Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [1,434] milliseconds.

log4j:WARN No appenders could be found for logger (org.apache.tapestry5.ioc.RegistryBuilder).

log4j:WARN Please initialize the log4j system properly.

14-Feb-2017 19:39:59.489 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive /var/lib/dfactor-apache-tomcat-8.5.11/webapps/wizard.war has finished in 3,713 ms

14-Feb-2017 19:39:59.491 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory /var/lib/dfactor-apache-tomcat-8.5.11/webapps/ROOT

14-Feb-2017 19:39:59.553 INFO [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.

14-Feb-2017 19:39:59.555 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory /var/lib/dfactor-apache-tomcat-8.5.11/webapps/ROOT has finished in 64 ms

14-Feb-2017 19:39:59.562 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler [http-nio-8080]

14-Feb-2017 19:39:59.567 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 3829 ms
```

Item 25 – Tomcat deployment – catalina.out

2.7 Verify access to wizard pages

Before you can verify wizard page, you need to ensure the wizard.keyscaler-672-001.com domain is resolvable. Note the domain will be different for your installation. This can be done using an active DNS server or adding an entry to your local hosts file, e.g. as follows:

```
35.166.122.161 wizard.keyscaler-672-001.com
```

Item 26 – Hosts file entry example

> ⚠ Note: If wizard deployed in a cloud infrastructure (e.g. Azure or AWS) you may need to configure Networking / security groups to allow for HTTP/S access.

Enter the following URL https://wizard.keyscaler-672-001.com in your browser and verify you can reach the wizard page.

3 Next Steps

Install KeyScaler 6.7.2 by following the steps outlined in [1].

----------End of Document----------

APPENDIX XX

Next, set the following permissions in tomcat:

```
$ ls -la /var/lib/dfactor-apache-tomcat-8.5.11
```

Item 8.2 – List files in tomcat directory

```
$ chown -R dfactor_user:tomcat /var/lib/dfactor-apache-tomcat-8.5.11
$ cd /var/lib/dfactor-apache-tomcat-8.5.11
$ chown -R dfactor_user:tomcat bin/ lib/ logs/ work/ webapps/ work/
$ chmod g+w bin
$ chmod 755 bin/
$ cd /var
$ chmod 755 www
$ chmod 755 dfactor/
$ chmod 755 dfactor/data/
$ chmod 755 /var/www/tomcat/bin
$ chmod 755 /var/www/tomcat/bin/dfactor
$ cd /var/www/tomcat/bin/dfactor
$ chmod 755 *.*
$ cd /var/lib/
$ chmod 755 dfactor-apache-tomcat-8.5.11/
```

Item 8.3 – Set tomcat Permissions

Test if you can start the dfactor service.

```
sh -x /etc/init.d/dfactor start
sh -x /etc/init.d/dfactor restart
```

Item 8.4 – Test starting dfactor services