Deploy CAST Security Dashboard using ZIP file

ⓘ **Summary:** This section describes how to **install** and **configure** the **Security Dashboard 2.x** in **ZIP** file format (i.e. no web application server required).

Introduction

What is the ZIP file format?

From release 2.0 onwards, CAST will deliver a **ZIP file for each CAST Dashboard**, alongside the traditional WAR file that has always been delivered. The ZIP file is a new method of deploying the CAST Dashboards based on **Spring Boot** and does not require a web application server (the application server is embedded in the ZIP itself). The aim of the ZIP file releases is to simplify and speed up the deployment of the CAST Dashboards. The deployment and configuration of the Spring Boot based dashboards differs slightly to the steps required for traditional WAR files (see **Deploy CAST Security Dashboard using WAR file**).

How many Security Dashboards do I need?

CAST highly recommends that you install **one** Security Dashboard and consolidate **all your CAST Dashboard Services schemas** into this.

Requirements

Please see **Standalone dashboard - installation requirements**, specifically the requirements for the **Engineering Dashboard**, which are identical.

Pre-installation check list

Before beginning the installation process, please ensure that you have carried out all of the following tasks and that the following requirements have been met:

✓	Ensure you have read all Release Notes accompanying CAST products for any last-minute information.
✓	Decide where the Security Dashboard will be installed and run from.
✓	Ensure that your user login on the target machine has sufficient user privileges to install applications.
✓	Make sure you have the required .ZIP file ready for deployment.

Installation procedure

The installation process is divided into various steps:

Step 1	Unpack the ZIP file

Step 2	Configure the connection parameters
Step 3	Install the license key
Step 4	Install the Microsoft Windows Service to start/stop the application server - optional
Step 5	Start the dashboard and test connection
Step 6	Configure user authentication
Step 7	First login and become admin - 2.1 only
Step 8	Configure roles
Step 9	Generate snapshot data for display
Step 10	Configure data authorization

Step 1 - Unpack the ZIP file

Unpack the ZIP file on the server. You can run the dashboard directly from wherever you unpack the ZIP file, however, you may want to move the unpacked files to a more appropriate location.

Step 2 - Configure the connection parameters

Modify application.properties to define connection to CSS/PostgreSQL for AIP schemas

When the ZIP has been unpacked you now need to configure the **application.properties** file to tell the web application on which **CAST Storage Service/PostgreSQL instance** the Dashboard schemas are stored. This file is located here:

```
<unpacked_zip>\configurations\application.properties
```

Locate the following section in the file:

```
## DATASOURCE
# Resource1 is the datasource name used in domains.properties
# Adapt server name (localhost) and port (2282) if required
# You can add multiple datasources if you want to connect to multiple CSS Servers. Datasource name must be
unique
# You have to configure your domains names and relative schema names in domains.properties
restapi.datasource[0].url=jdbc:postgresql://localhost:2282/postgres
restapi.datasource[0].username=operator
restapi.datasource[0].password=CastAIP
restapi.datasource[0].poolname=Resource1
restapi.datasource[0].minimumIdle=10
restapi.datasource[0].maximumPoolSize=20
```

If all your Dashboard schemas are located **on one single CAST Storage Service/PostgreSQL instance** then you need to modify the `url`, `username` and `password` entries to match your target CAST Storage Service/PostgreSQL, for example:

```
## DATASOURCE
# Resource1 is the datasource name used in domains.properties
# Adapt server name (localhost) and port (2282) if required
# You can add multiple datasources if you want to connect to multiple CSS Servers. Datasource name must be
unique
# You have to configure your domains names and relative schema names in domains.properties
restapi.datasource[0].url=jdbc:postgresql://192.168.200.104:2282/postgres
restapi.datasource[0].username=operator
restapi.datasource[0].password=CastAIP
restapi.datasource[0].poolname=Resource1
restapi.datasource[0].minimumIdle=10
restapi.datasource[0].maximumPoolSize=20
```

If your Dashboard schemas are located **on multiple CAST Storage Services/PostgreSQL instances**, you need to add in the additional servers as shown in the example below:

- Ensure that you modify the `url`, `username, password` and `resource` entries to match your target CAST Storage Service/PostgreSQL. In particular, the `resource` entry must be unique within the **application.properties** file.
- The `[0]` must also be incremented for additional CAST Storage Service/PostgreSQL instances, for example, use `restapi.datasource[1]`, `restapi.datasource[2]` etc.

```
## DATASOURCE
# Resource1 is the datasource name used in domains.properties
# Adapt server name (localhost) and port (2282) if required
# You can add multiple datasources if you want to connect to multiple CSS Servers. Datasource name must be unique
# You have to configure your domains names and relative schema names in domains.properties
restapi.datasource[0].url=jdbc:postgresql://192.168.200.104:2282/postgres
restapi.datasource[0].username=operator
restapi.datasource[0].password=CastAIP
restapi.datasource[0].poolname=Resource1
restapi.datasource[0].minimumIdle=10
restapi.datasource[0].maximumPoolSize=20

restapi.datasource[1].url=jdbc:postgresql://192.168.200.105:2282/postgres
restapi.datasource[1].username=operator
restapi.datasource[1].password=CastAIP
restapi.datasource[1].poolname=Resource2
restapi.datasource[0].minimumIdle=10
restapi.datasource[0].maximumPoolSize=20
```

Save the file before proceeding.

minimumIdle and maximumPoolSize

The following options are used to govern the connections from the web application to the target CAST Storage Service/PostgreSQL instance:

```
restapi.datasource[0].minimumIdle=10
restapi.datasource[0].maximumPoolSize=20
```

CAST recommends using the default options unless you are experiencing performance issues. The options are used as follows:

minimumIdle	The minimum number of connections that should be kept in the pool at all times (even if there is no traffic). Default value is 10. Idle connections are checked periodically.
maximumPoolSize	The maximum number of active connections that can be allocated from this pool at the same time. The default value is 20.

See also **Configure the Health Dashboard for large numbers of Applications**.

Modify domains.properties

You now need to configure the **domains.properties** file which provides a link between the **CAST Storage Services/PostgreSQL instances** defined in the **application.properties** file and the Dashboard schemas containing the relevant snapshot data. This file is located here:

```
<unpacked_zip>\configurations\domains.properties
```

This file is delivered empty as shown below:

```
# Domains for SD
# empty lines in this file lead to connection error, remove all empty lines
# - You have to align [Resource1] with the resource name configured in application.properties
# - You have to replace [Central Schema1] by the central schema name
# - Domains names must be unique
# AED1=Resource1,[Central Schema1]
# AED2=Resource1,[Central Schema2]
```

For each Dashboard schema that you need to display in the CAST Security Dashboard, **add one line to the file** ensuring that there are no empty lines:

AED1	This is known as the "domain" and this **must be unique** in the **domains.properties** file. Therefore for each Dashboard schema you need to display in the CAST Security Dashboard, you need to assign **one unique domain**. You can use any domain name notation you want, however, CAST highly recommends incrementing the number, i.e. AED1, AED2, AED3 etc.
Resource1	This entry refers to the CAST Storage Service/PostgreSQL instance as defined in the **application.properties** file.
[Central Schema1]	This entry refers to the Dashboard schema containing the relevant Application data.

For example, for one single Dashboard schema called "MEUDON_CENTRAL" stored in the CAST Storage Service/PostgreSQL instance defined in **Resource1** in the **application.properties** file, add the following:

```
# Domains for SD
# empty lines in this file lead to connection error, remove all empty lines
# - You have to align [Resource1] with the resource name configured in application.properties
# - You have to replace [Central Schema1] by the central schema name
# - Domains names must be unique
# AED1=Resource1,[Central Schema1]
# AED2=Resource1,[Central Schema2]
AED1=Resource1,MEUDON_CENTRAL
```

For multiple Dashboard schemas where all schemas are located in the same CAST Storage Service/PostgreSQL instance defined in **Resource1** in the **application.properties**, add the following:

```
# Domains for ED
# empty lines in this file lead to connection error, remove all empty lines
# - You have to align [Resource1] with the resource name configured in application.properties
# - You have to replace [Central Schema1] by the central schema name
# - Domains names must be unique
# AED1=Resource1,[Central Schema1]
# AED2=Resource1,[Central Schema2]
AED1=Resource1,MEUDON_CENTRAL
AED2=Resource1,SEVRES_CENTRAL
AED3=Resource1,PARIS_CENTRAL
```

For multiple Dashboard schemas where the schemas are located on different CAST Storage Services/PostgreSQL instances (**Resource1** and **Resource2**) as defined in the **application.properties** file, add the following:

```
# Domains for SD
# empty lines in this file lead to connection error, remove all empty lines
# - You have to align [Resource1] with the resource name configured in application.properties
# - You have to replace [Central Schema1] by the central schema name
# - Domains names must be unique
# AED1=Resource1,[Central Schema1]
# AED2=Resource1,[Central Schema2]
AED1=Resource1,MEUDON_CENTRAL
AED2=Resource2,SEVRES_CENTRAL
AED3=Resource2,PARIS_CENTRAL
```

Save the file before proceeding.

Modify application.properties to define connection to CSS/PostgreSQL for the roles/permissions schema - 2.1 only

In **2.1 only**, an interface exists to manage **User roles - 2.x and above** and **Data authorization - 2.x and above** - this interface stores all its records in a **dedicated schema** on a **CAST Storage Service/PostgreSQL instance**. This instance does not need to be the same as used for your AIP schemas (Dashboard/Measure schemas), however, the required schema is small and therefore CAST recommends using an existing CAST Storage Service/PostgreSQL instance to host it. The schema is created automatically when you start up your Dashboard deployment if it does not already exist.

The **application.properties** file contains a section dedicated to this schema - this file is located here:

```
<unpacked_zip>\configurations\application.properties
```

Locate the following section in the file:

```
#datasource configuration for user management
spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?currentSchema=cast_dashboards
spring.datasource.platform=postgres
spring.datasource.username=operator
spring.datasource.password=CastAIP
spring.datasource.initialization-mode=always
spring.datasource.driver-class-name=org.postgresql.Driver
spring.liquibase.change-log=classpath:db/changelog/db.changelog-master.xml
spring.liquibase.default-schema=cast_dashboards
spring.liquibase.enabled=true
```

Change the line `spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?currentSchema=cast_dashboards` to match the CAST Storage Service/PostgreSQL instance you intend to use for the roles/permissions schema, for example:

```
#datasource configuration for user management
spring.datasource.url=jdbc:postgresql://192.168.200.104:2282/postgres?currentSchema=cast_dashboards
spring.datasource.platform=postgres
spring.datasource.username=operator
spring.datasource.password=CastAIP
spring.datasource.initialization-mode=always
spring.datasource.driver-class-name=org.postgresql.Driver
spring.liquibase.change-log=classpath:db/changelog/db.changelog-master.xml
spring.liquibase.default-schema=cast_dashboards
spring.liquibase.enabled=true
```

Save the file before proceeding. This will ensure that a schema called "**cast_dashboards**" is created on the target CAST Storage Service /PostgreSQL instance when you start the web application.

Step 3 - Install the license key

As explained in **Dashboard Service license key configuration**, when you want to access a **Dashboard schema** using the **CAST RestAPI** (i.e. via the **Security Dashboard**, or via the **CAST Report Generator**), a special **license key** is required. This license key grants specific access to one or multiple Dashboard schemas for the web application in which it is installed (i.e. the Engineering Dashboard or the CAST RestAPI).

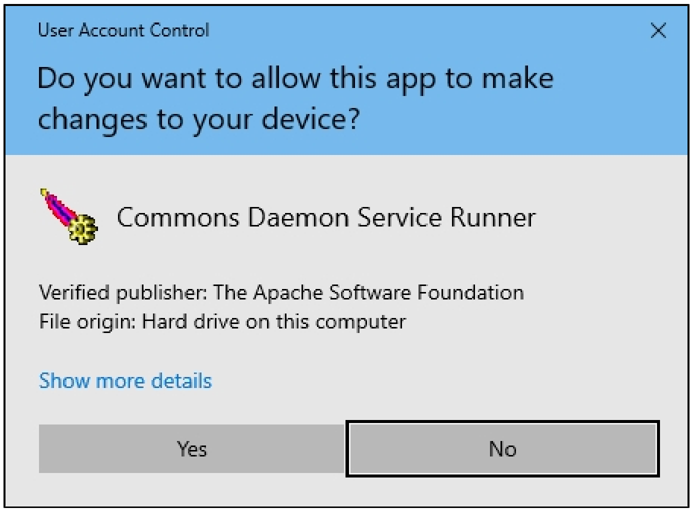
You must therefore install the license key and, if you are using a restricted license key, define data access authorization. These two steps are explained in **Dashboard Service license key configuration** in the sections **How do I install a license key?** and **How to authorize users when using a RESTRICTED license key**.

Step 4 - Install the Microsoft Windows Service to start/stop the application server - optional

If you have deployed the ZIP file on Microsoft Windows and would like to control the application server via a **Microsoft Windows Service**, CAST provides an installation batch script to do this for you. Locate the following file:

```
<unpacked_zip>\dashboard-service-install.bat
```

Double click this file to start the service installation. You may be prompted to accept a UAC warning:



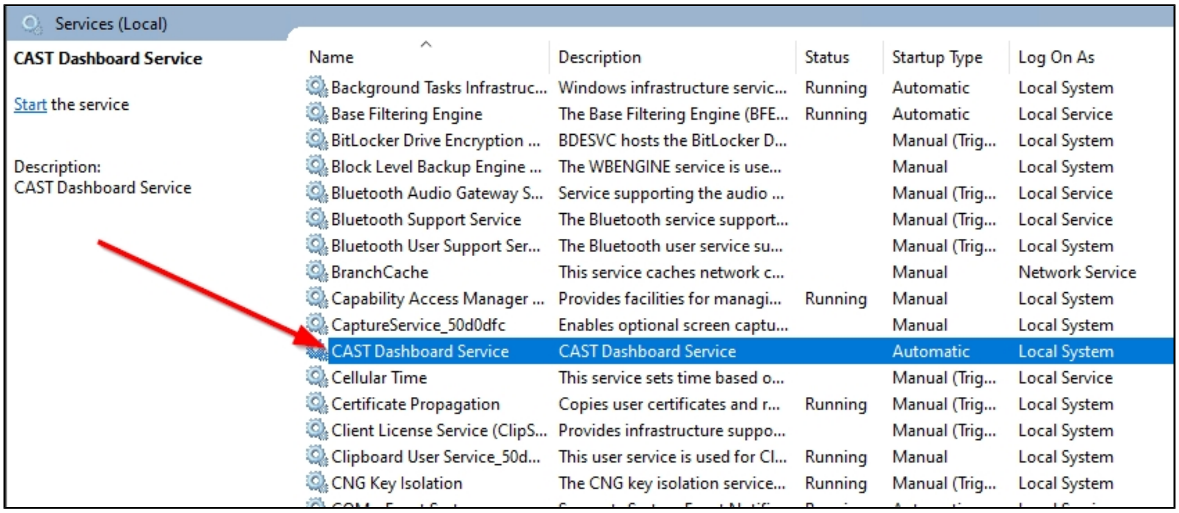
User Account Control

Do you want to allow this app to make changes to your device?

Commons Daemon Service Runner

Verified publisher: The Apache Software Foundation
File origin: Hard drive on this computer

On completion the service will be listed as **CAST Dashboard Service** with a startup type set to **Automatic**, log on as **Local System** and will not be running:



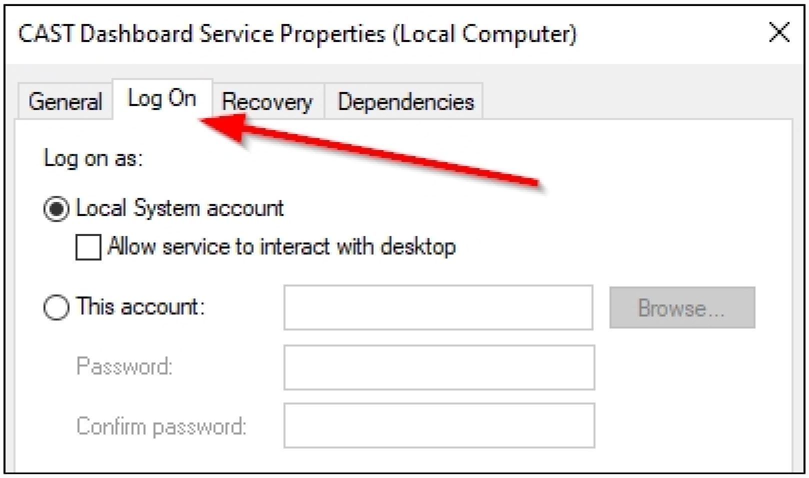
CAST Dashboard Service

Start the service

Description:
CAST Dashboard Service

Name	Description	Status	Startup Type	Log On As
Background Tasks Infrastruc...	Windows infrastructure servic...	Running	Automatic	Local System
Base Filtering Engine	The Base Filtering Engine (BFE...	Running	Automatic	Local Service
BitLocker Drive Encryption ...	BDESVC hosts the BitLocker D...		Manual (Trig...	Local System
Block Level Backup Engine ...	The WBENGINE service is use...		Manual	Local System
Bluetooth Audio Gateway S...	Service supporting the audio ...		Manual (Trig...	Local Service
Bluetooth Support Service	The Bluetooth service support...		Manual (Trig...	Local Service
Bluetooth User Support Ser...	The Bluetooth user service su...		Manual (Trig...	Local System
BranchCache	This service caches network c...		Manual	Network Service
Capability Access Manager ...	Provides facilities for managi...	Running	Manual	Local System
CaptureService_50d0dfc	Enables optional screen captu...		Manual	Local System
CAST Dashboard Service	CAST Dashboard Service		Automatic	Local System
Cellular Time	This service sets time based o...		Manual (Trig...	Local Service
Certificate Propagation	Copies user certificates and r...	Running	Manual (Trig...	Local System
Client License Service (ClipS...	Provides infrastructure suppo...		Manual (Trig...	Local System
Clipboard User Service_50d...	This user service is used for Cl...	Running	Manual	Local System
CNG Key Isolation	The CNG key isolation service...	Running	Manual (Trig...	Local System

- The installer requires:
  - **Java JDK/JRE 8**
  - a **JAVA_HOME** system environment variable pointing to the installation location of the Java JDK
- The .bat installer will configure the service to use the **<unpacked_zip>\amd64\dashboard-service.exe**. You may want to ensure that the unpacked zip file is in an appropriate location.
- You can change the log on as, after the install has completed by right clicking the service and changing the options in the **Log On** tab:



CAST Dashboard Service Properties (Local Computer)

Log on as:

● Local System account

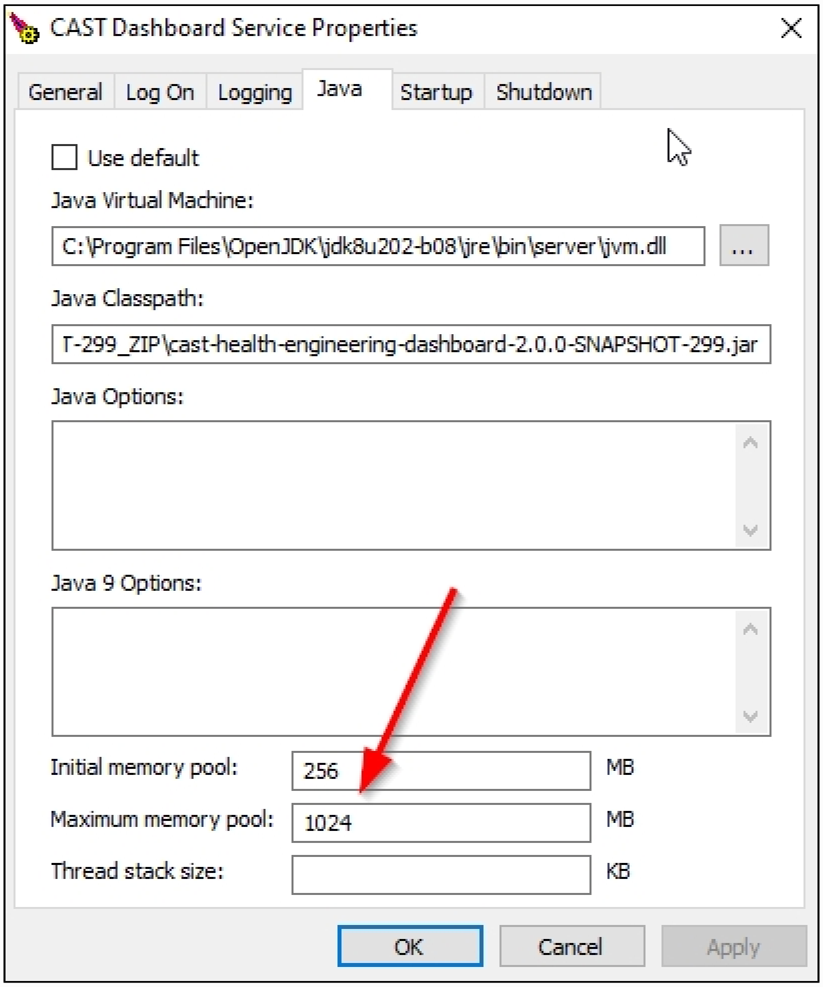
□ Allow service to interact with desktop

○ This account:

Password:

Confirm password:

- The installer will set the service to use the following RAM memory - you may find that this is not sufficient. See Apache Tomcat performance considerations for information about memory requirements.
  - Initial memory pool = 256MB
  - Maximum memory pool = 1024MB



CAST Dashboard Service Properties

□ Use default

Java Virtual Machine: C:\Program Files\OpenJDK\jdk8u202-b08\jre\bin\server\jvm.dll

Java Classpath: T-299_ZIP\cast-health-engineering-dashboard-2.0.0-SNAPSHOT-299.jar

Java Options:

Java 9 Options:

Initial memory pool: 256 MB

Maximum memory pool: 1024 MB

Thread stack size: KB

Step 5 - Start the dashboard and test connection

To start the dashboard:

- either **start the Windows Service** if you are using Microsoft Windows and have chosen to install the Windows Service
- or **run the following file:**

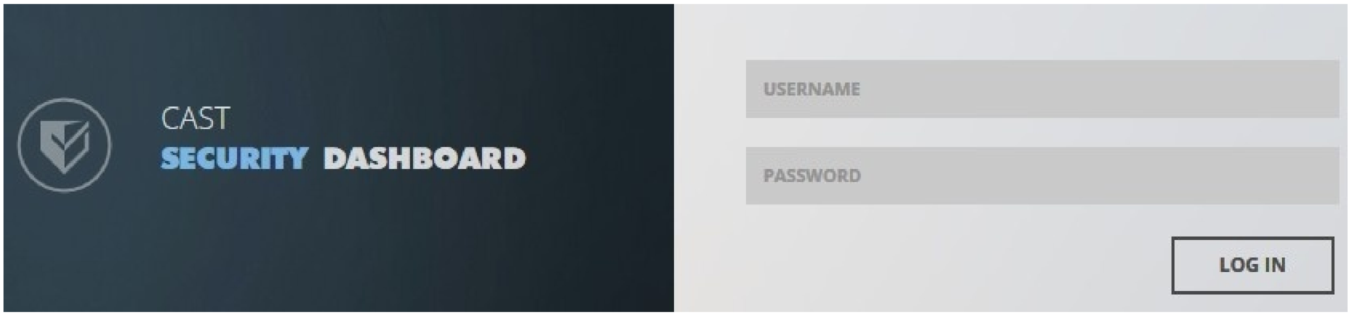
```
Microsoft Windows
<unpacked_zip>\startup.bat

Linux
<unpacked_zip>\startup.sh
Note that you may need to run "chmod +x startup.sh" to grant execution permission to the script before running
it.
You may also need to run this file with elevated permissions (e.g. sudo) using the following syntax "./startup.
sh"
```

By default the dashboard is configured to run on **port 8080** (this can be changed, see **Modify the user access port for 2.x JAR or ZIP deployments**). Use the following URL - where **<server_name>** is equal to the host name of the current server to access the dashboard. If you are testing on the server itself, you can use `http://localhost:8080`:

```
http://<server_name>:8080
```

You should see the login page as follows - this indicates that the initial setup was successful:



Error messages are documented in **Error Messages**.

Step 6 - Configure user authentication

This step involves configuring how your users will **authenticate** with the CAST Security Dashboard. Most organizations opt for LDAP/Active Directory integration so that users can use their corporate username/password to access the resources they need. The Security Dashboard also has a built in username/password authentication mechanism which is enabled "out of the box". See **User authentication**.

Step 7 - First login and become admin - 2.1 only

By default, the CAST Dashboard requires that at least one user is granted the **ADMIN role** following the first login after the **User authentication** configuration. This ensures that one user can access all data and cofiguration settings. See **First login and become admin**. This step is not required when using **Dashboards 1.x** and can be skipped.

Step 8 - Configure roles

This step involves configuring **roles** for users and groups that are accessing the CAST Security Dashboard. See **User roles**.

Step 9 - Generate snapshot data for display

Before your users can "consume" data via the CAST Security Dashboard, you need to **generate snapshot data**.

Step 10 - Configure data authorization

An **Authorization** defines permission to access and "consume the data" in **a specific Application** or **group of Applications** via the CAST Security Dashboard. If permission is not granted, or a "restriction" is used, then any information related to this Application will be not accessible: application properties such as name, technologies or grades and measures, etc. Therefore, an Authorization must be defined before a user/group of users can access a specific application. See **Data authorization**.

Additional information

Advanced configuration specific to the CAST Engineering/Security Dashboard (the two are identical in terms of configuration):

- CAST Dashboard Service schema connection configuration pooling
- Education - change Share and Promote email text
- Engineering Dashboard json configuration options
- Engineering Dashboard tile management
    - Top Modules with Violations or Critical Violations tile
    - Health Factor weakness or strength tile
    - Risk Introduced tile
    - Top Rules with increasing or decreasing violations tile
    - Action Plan tile
    - Exclusions tile
    - External Links tile
    - Background Facts tile
    - Custom tile
    - Industry Standard tiles
    - OMG Technical Debt tile
    - Architecture Model violation tile
    - Top Riskiest Components tile
    - Top Riskiest Transactions tile
- Integration of CAST Action Plan into Atlassian JIRA
- Managing the Engineering Dashboard search indexes
- Report Generation configuration
- Using the GUI and CLI tools for Engineering Dashboard
- View custom Business Criteria in Engineering Dashboard

Additional advanced configuration options:

- For 2.x JARs or ZIPs
    - Changing the name of the cast_dashboards schema
    - Configure additional Dashboard schemas for JAR file deployments
    - Deploying multiple 2.x ZIPs or JARs on the same server
    - Modify the user access port for 2.x JAR or ZIP deployments
    - Start and stop 2.x JAR or ZIP deployments
- Configuring the Log and Audit Trail
- Dashboard localization
- Encrypt login and password for database and LDAP
- Error Messages
- General timeout setting
- Injecting custom tags
- Lost password and request access configuration
- Modifying default dashboard text
- Modifying login error messages
- Multiple CAST dashboard installation scenario
- Reload the cache
- Removing snapshot data
- RestAPI authentication using an API key