Cognizant

CIQDashboard

INSTALLATION GUIDE

WINDOWS OS

Version 3.1

Contents

Legal Notice

These materials have not been published and are not available to the general public because they are proprietary to Cognizant and contain Cognizant confidential and/or trade secret information, including without limitation the descriptions of features, functions, methods, techniques and technical concepts contained herein. These materials may also contain information covered by U.S. and/or foreign patents or patent applications. Use, distribution or copying of these materials, in whole or in part, is forbidden, except with Cognizant's express prior written permission.

The documentation is provided "as is" and all express or implied conditions, representations and warranties, including any implied warranty of merchantability, fitness for a particular purpose or non-infringement, are disclaimed, except to the extent that such disclaimers are held to be legally invalid. Cognizant shall not be liable for any incidental or consequential damages in connection with the furnishing, performance, or use of this documentation. The information contained in this documentation is subject to change without notice.

Cognizant® and Cognizant® CIQDashboard are either registered trademarks or trademarks of Cognizant in the United States and other countries. Other trademarks referenced herein are the property of their respective owners.

About this Installation Guide

The Cognizant® CIQDashboard Installation Guide, provides help to install CIQDashboard Server in your system. It guides you through the steps to install and uninstall the CIQDashbaord software, and provides instructions for completing the minimal configuration required to start creating dashboards. In addition, it provides troubleshooting information during or post-installation of the software.

The installation guide consists of the following chapters:

Chapter Name	Description
About Cognizant® CIQDashboard	Provides information about CIQDashboard
Hardware Requirements	Provides an overview about required hardware configuration
Software Requirements	Provides an overview about prerequisites for installing Cognizant CIQDashboard
Binaries/Setup Files	Provides information on the required binaries/setup files
Installation Procedure in Windows OS	Provides procedure to install Cognizant CIQDashboard
Uninstallation Procedure	Provides procedure to uninstall Cognizant CIQDashboard
FAQs	Provides answers to a list of commonly asked questions regarding Cognizant CIQDashboard

About Cognizant® CIQDashboard

This chapter consists:

Product Overview

Intelligent Dashboard (CIQDashboard) is a data visualization solution, designed to transform data reporting into interactive business intelligence dashboards.

Hardware Requirements

The following table lists the hardware requirements for CIQDashboard:

Type	Description
Operating System	Windows XP and above
Processor	64-bit multi-core
RAM	Minimum: 8 GB; Recommended: 16 GB
HDD	100 GB of available space (can increase based on volume of data gathered from the client)
Monitor	Resolution of 1024x768 or greater

Software Requirements

The following table lists the software prerequisites for CIQDashboard and in a Windows Operating System:

Software	Download Link
Java JDK	https://www.oracle.com/java/technologies/javase-jdk11-downloads.html Required version: 11
NGINX	http://nginx.org/en/download.html (Stable) Required version: 1.18 or above
MongoDB	https://www.mongodb.com/try/download/community Required version: 4.0 or above
Robo 3T	https://robomongo.org/download Required version: 1.4 or above

Cognizant® strongly recommends that the MongoDB instance provided by the customer should be configured to enable encryption at rest. Please refer, https://docs.mongodb.com/manual/core/security-encryption-at-rest/ for more information.

Admin or Power Broker privileges are required to install MongoDB and run Nginx Server

Binaries/Setup Files

1. Please download the binaries from CIQDashboard Team. You can reach the **CIQDashboard Team** for latest binaries/setup files.
2. Create a new folder in a drive (**Example:** C:\ciqdashboard\ciqdashboard_deployment\binaries) or use any existing folder

The path and the folder names mentioned, are only an example and are not mandatory to be the same

Installation Procedure in Windows OS

This chapter provides procedural information for installing the CIQDashbaord application in Windows OS.

Mongo DB Configuration

This section describes the steps to install Mongo DB and set up the server.

1.1.1. Installing Mongo DB

Follow the steps below after downloading the software. Refer the section Software Prerequisites for downloading Mongo DB.

1. In Windows Explorer, locate the downloaded **MongoDB .msi** file.
2. Double-click the **.msi** file. A set of screens appear to guide you through the installation process

1.1.2. Authenticating Mongo DB

Follow the steps below to authenticate Mongo DB server.

Start Mongo Server in Normal Mode:

1. Create a folder with name: **mongoDB_Data**** in **C:\'**.
2. Open the command prompt (run as administrator), type **C:\Program Files\MongoDB\Server\4.0\bin** and press **Enter**. The folder path opens in the Command Prompt windows.
3. In the command prompt, type **mongod --dbpath c:\mongoDB_Data** and press **Enter**. The Mongo server starts in normal mode.

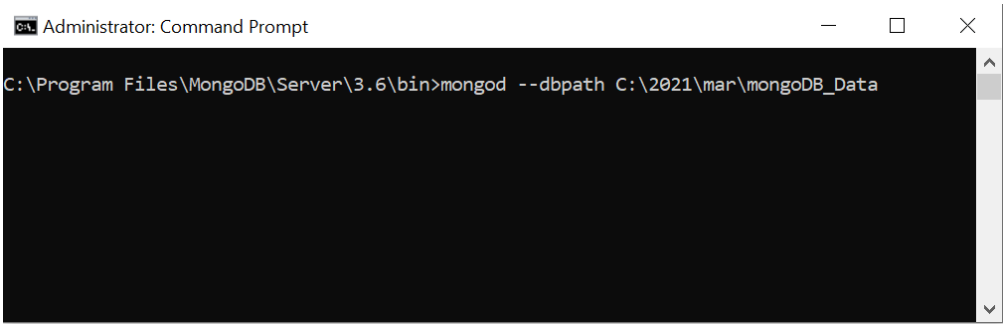

Figure 1: Administrator Command Prompt

Note	The path and the folder names mentioned, are only an example and are not mandatory to be the same

Open Mongo Shell:

In the Command Prompt, navigate to the **bin** folder of the Mongo installation, type **mongo** and press **Enter**. The Mongo shell opens in the same command prompt.

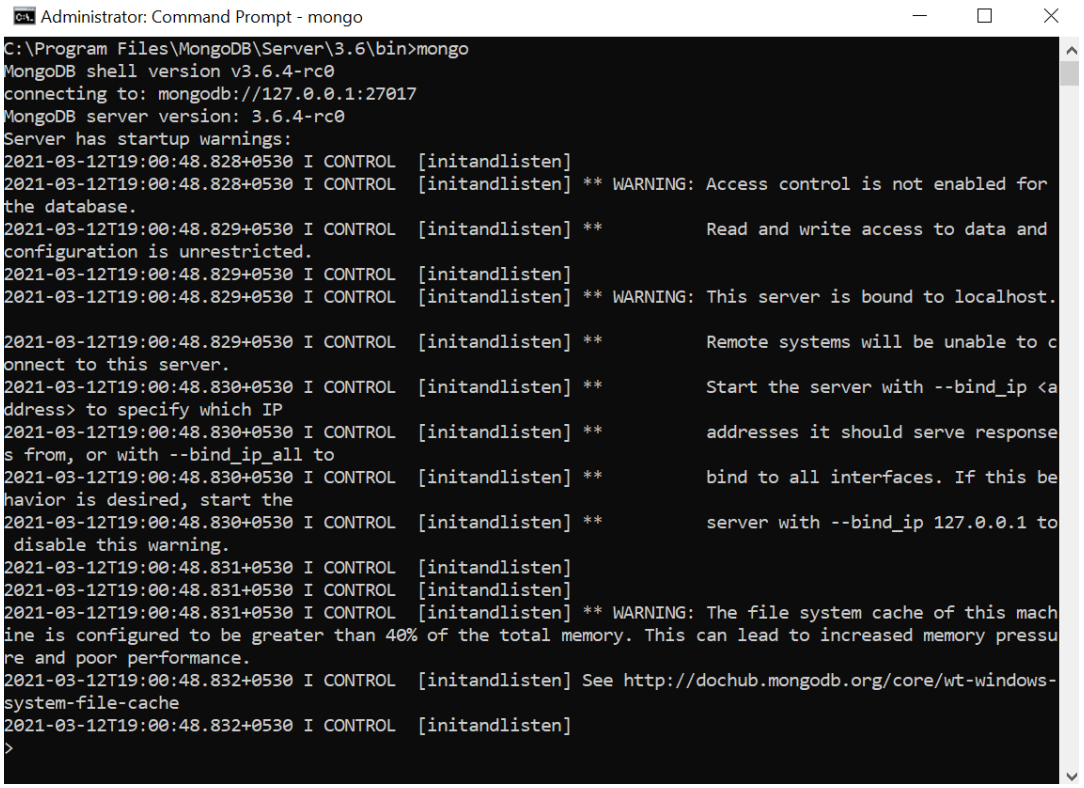

Figure 2: Mongo

Creating Admin User and Password:

1. In the mongo command prompt, type **use admin**. The DB switches to admin

Figure 3: use admin

2. To create admin user credentials and add it to the DB, type the command as:

   db.createUser({ user: "admin", pwd: "adminpassword", roles: [{ role: "root", db: "admin" }]})

Figure 4: Admin user

3. To check if the user credential is authenticated, in the mongo command prompt, type: **db.auth("admin","adminpassword")**. The command returns with value 1 for successful authentication.

Create Custom DB and its Users:

Follow the steps below in mongo command prompt to create custom DB and its users.

1. Type the command: **use ciqdasboard_prod**. The DB switches to c**iqdashboard_prod**.
2. Type the command: **db.sample.save({username:"root"})**
3. Type the command: **db.createUser({ user:"ciqdashboard", pwd:"ciqdashboard", roles: [{ role:"readWrite", db: "ciqdashboard_prod" }] })**

Figure 5: ciqdashboard_prod

Note

Db, user, and password mentioned are only an example and are not mandatory to be the same.

1.1.3. Validating Admin DB in RoboMongo

The authenticated admin DB can be validated using **RoboMongo** tool, which is Mongo shell UI. To validate:

1. Open **Robo 3T – 1.4**. Click **File -> Manage Connections**. The **MongoDB Connections** pane appears

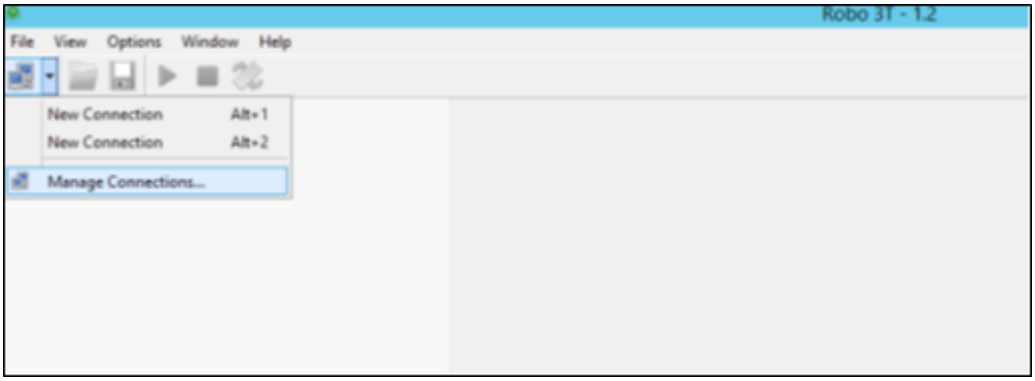

Figure 6: Manage connections

2. In **MongoDB Connections** pane, click **Create**. The **Connection Settings** pane appears.

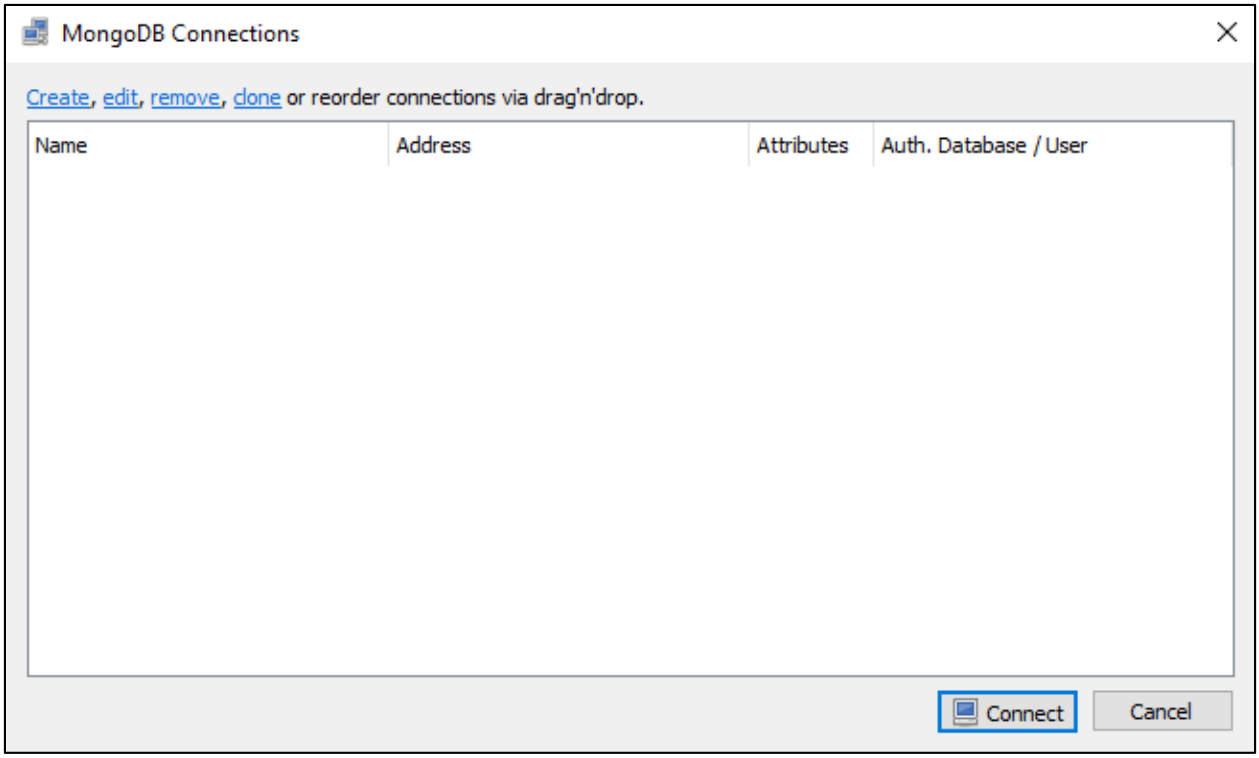

Figure 7: Create

3. In the **Connection Settings** pane, click **Connection** tab.

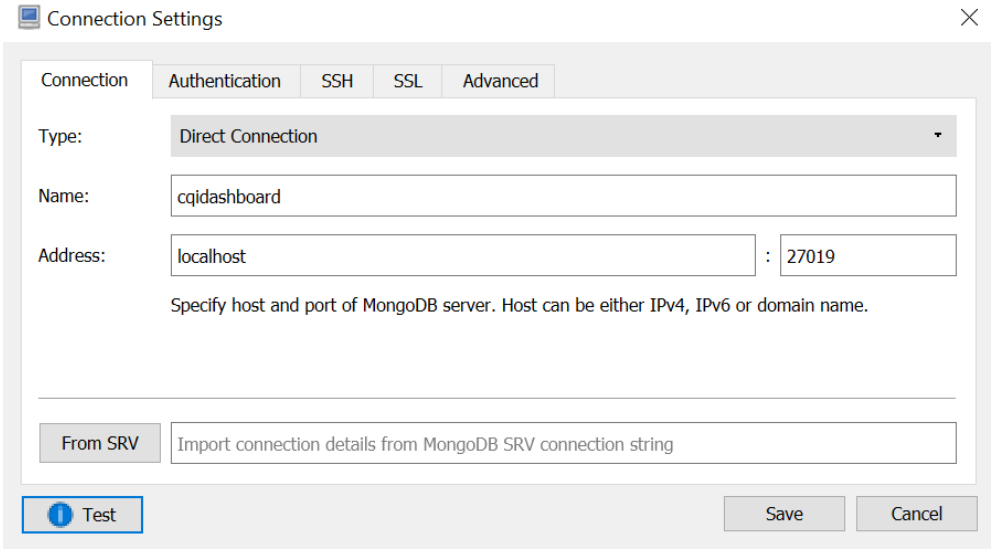

Figure 8: Connection

4. In **Name**, type a name for the connection you are creating.
5. In **Address**, type the host address and the port.
6. Click **Authentication** tab. The **Authentication** pane appears.

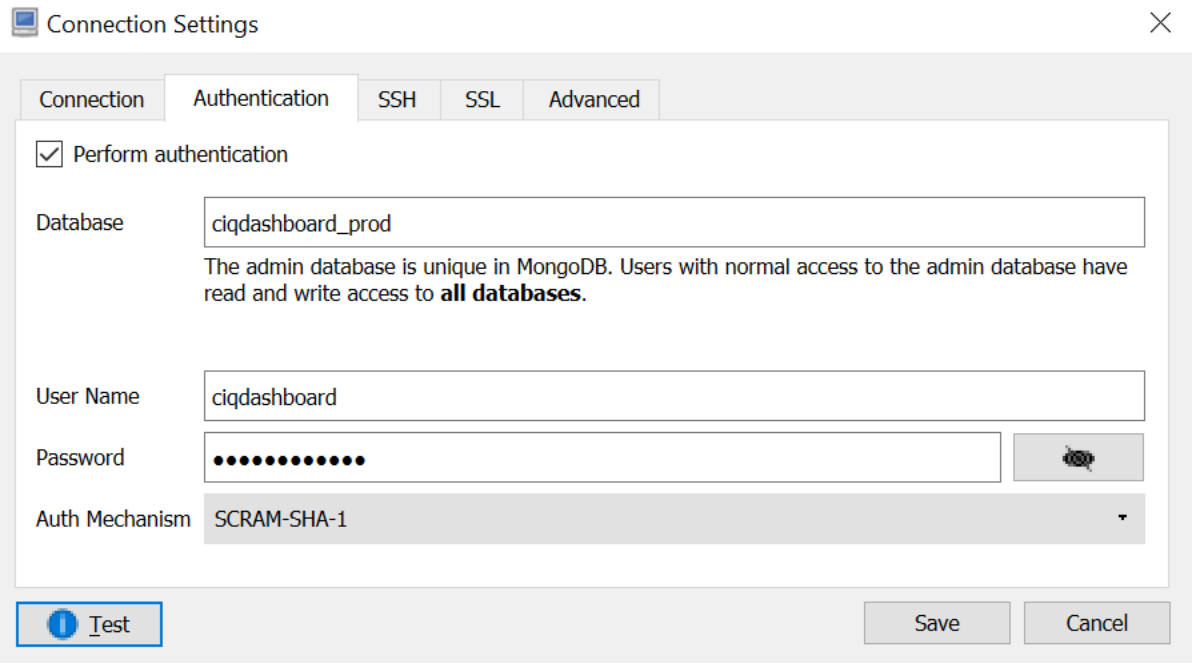

Figure 9: Authentication

7. In **Database**, type c**iqdashboard_prod** (db created in Create Custom DB and its Users)
8. In the **User Name** and **Password** fields, type the username and password created for admin database in Mongo shell.
9. From the **Auth Mechanism** drop-down, select **SCRAM-SHA-1**. The **Diagnostic** pop-up appears, displaying the connection and authorization status.

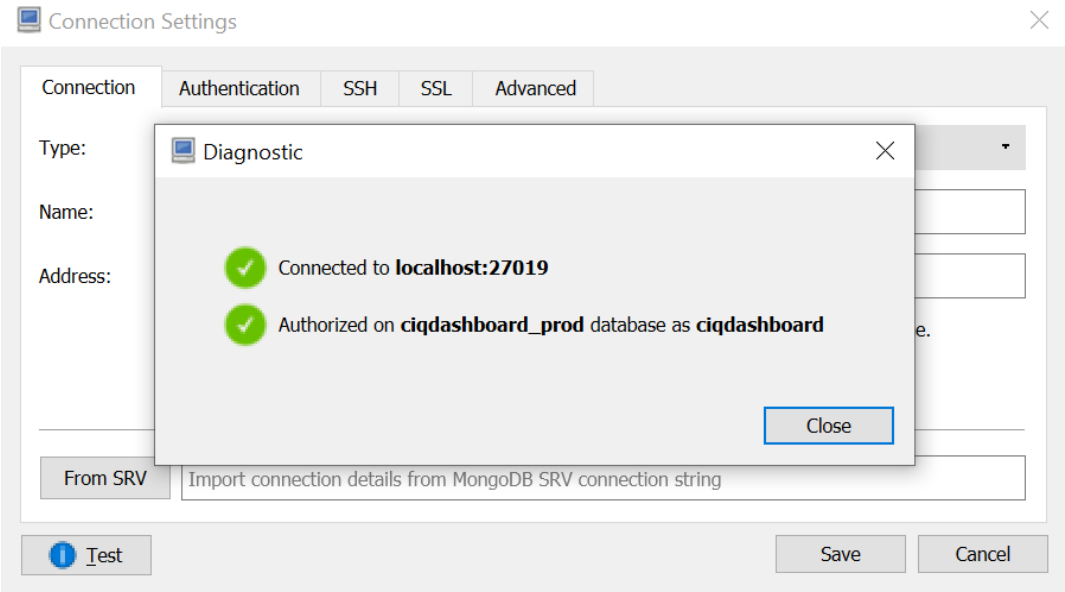

Figure 10: Diagnostic

1.1.4. Configuring MongoDB for CIQDashboard

1. Create a new folder in a drive (**Example:** C:\ciqdashboard\ciqdashboard_deployment\binaries)
2. In the created folder, place all the binaries (Refer the section Binaries/Setup Files)
3. Copy the content of c**iqdashboard-mongo-base-db.js** in **Robo 3T shell** in ciqdashboard database (db created in Create Custom DB and its Users)
4. Click **Execute**

Note	The path and the folder name mentioned, are only an example and are not mandatory to be the same

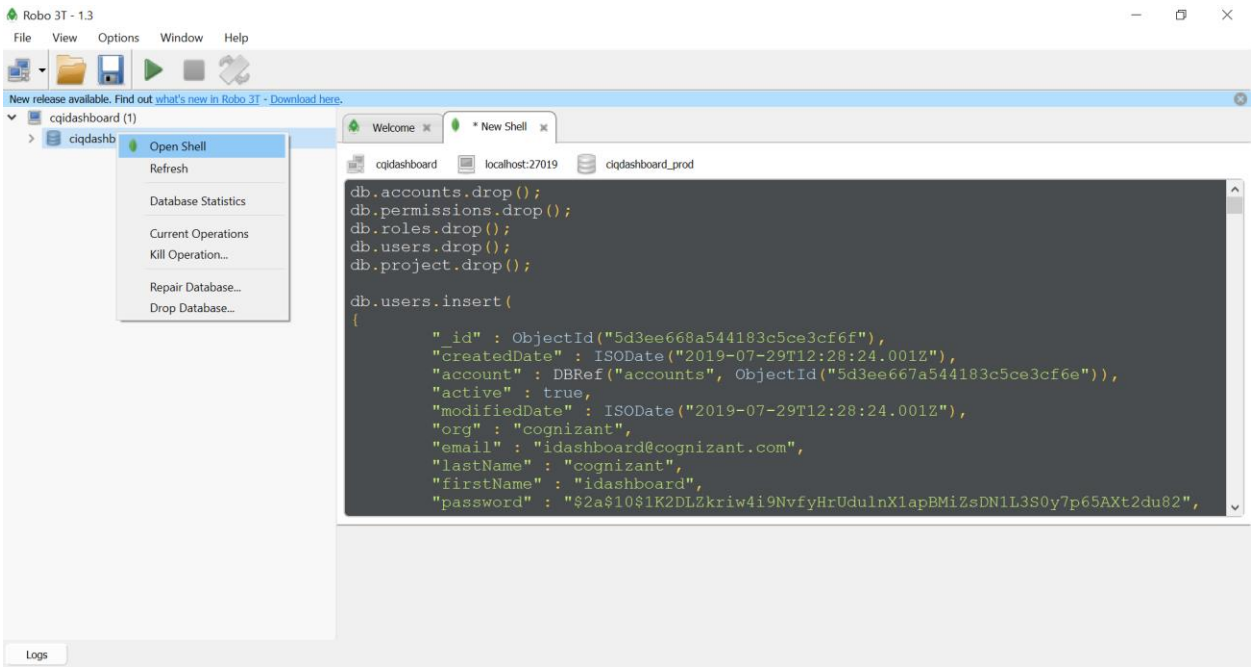

Figure 11: Execute

Note	To start the Mongo in Authentication Mode, stop Mongo opened in Normal mode and run the command: **mongo –auth–dbpath c:\mongoDB_Data**

Configuration of CIQDashboard

This section describes the steps to configure CIQDashboard.

1.1.5. Encrypt Password

1. The **jasypt-1.9.3-dist.zip** utility bundled with CIQDashboard binaries (Refer the section Binaries/Setup Files), encrypts the passwords

2. Unzip the **jasypt-1.9.3-dist.zip** file.
3. In command prompt navigate to the folder **jasypt-1.9.3**
4. Run the below command in command prompt (input =ciqdashboard, password=ciqdashboardSecurityKey):

   java -cp lib/jasypt-1.9.3.jar org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI password=ciqdashboardSecurityKey algorithm=PBEWITHHMACSHA512ANDAES_256 input=ciqdashboard ivGeneratorClassName=org.jasypt.iv.RandomIvGenerator

Note	Input is the password that requires encryption and password is the secret key.

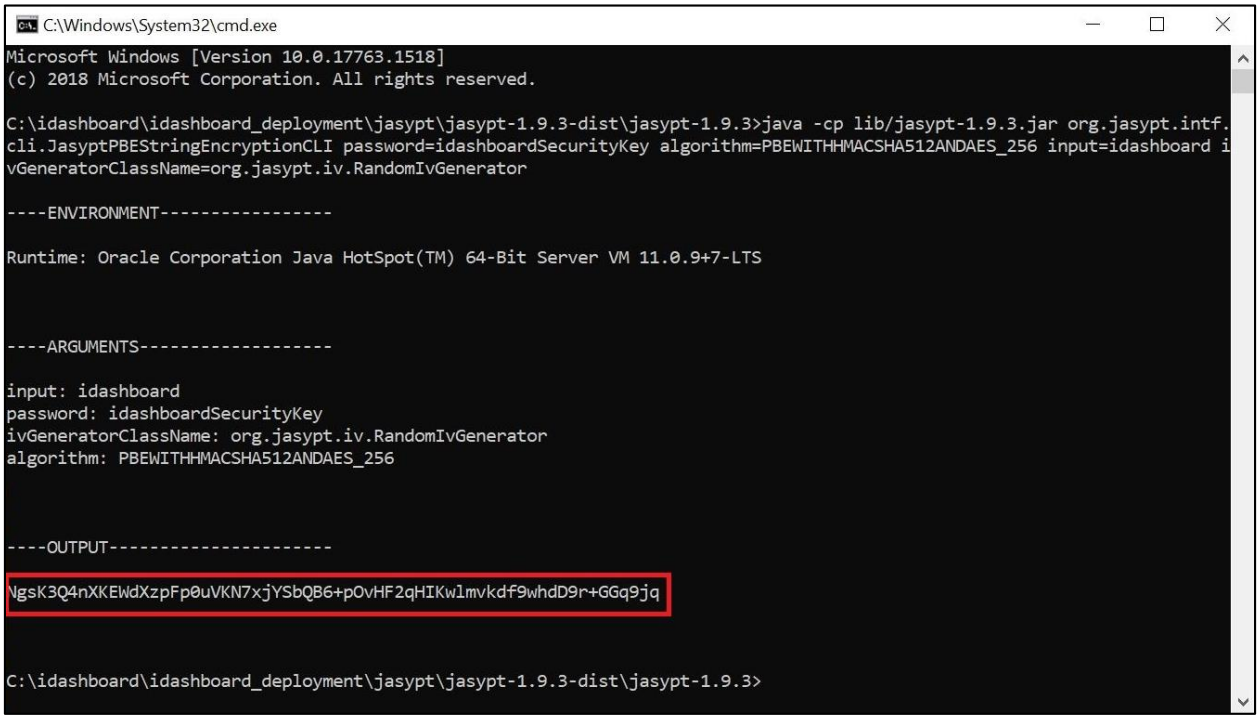

Figure 12: Output

5. The output is the encrypted key, copy and save the key.
6. Encode the security key (ciqdashboardSecurityKey - aWRhc2hib2FyZFNlY3VyaXR5S2V5) with **https://www.base64encode.org/** and pass it in the command line as **--jasypt.encryptor.password=<encoded_jasypt_pass>**

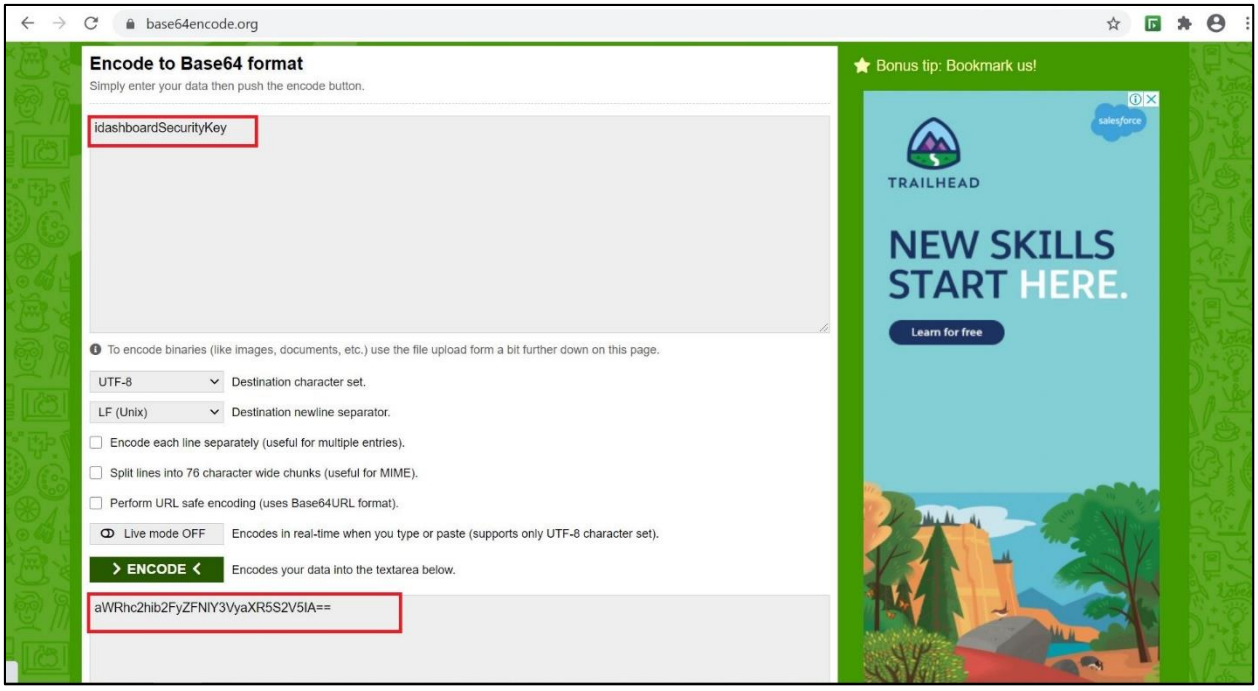

Figure 13: Encode

7. Pass generated encrypted key (application keys - mongodb, collector applications) in the command line. **Example:** ``--spring.data.mongodb.credents=ENC(<ecrypted_key>)`` as encypted value.

1.1.6. Execute ciqdashboard-api-0.0.1.jar

1. Open the command prompt and navigate to **bin** folder (Refer the section Binaries/Setup Files) java -jar ciqdashboard-api-0.0.1.jar --spring.data.mongodb.credents=**ENC(<ecrypted_key>)** --jasypt.encryptor.password=**<encoded_jasypt_pass>**

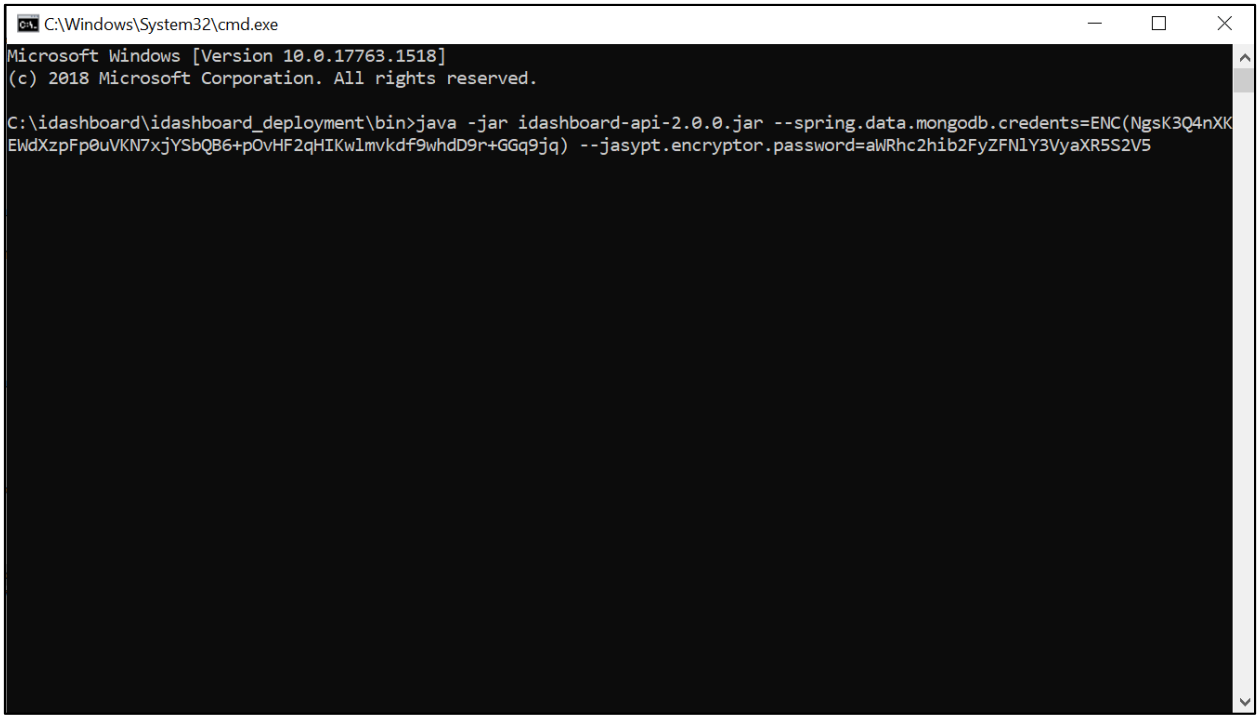

Figure 14: ciqdashboard api jar

2. Validate the port that c**iqdashboard-api** is listening (Refer the section NGINX Configuration,to validate the port in **nginx.conf** file)

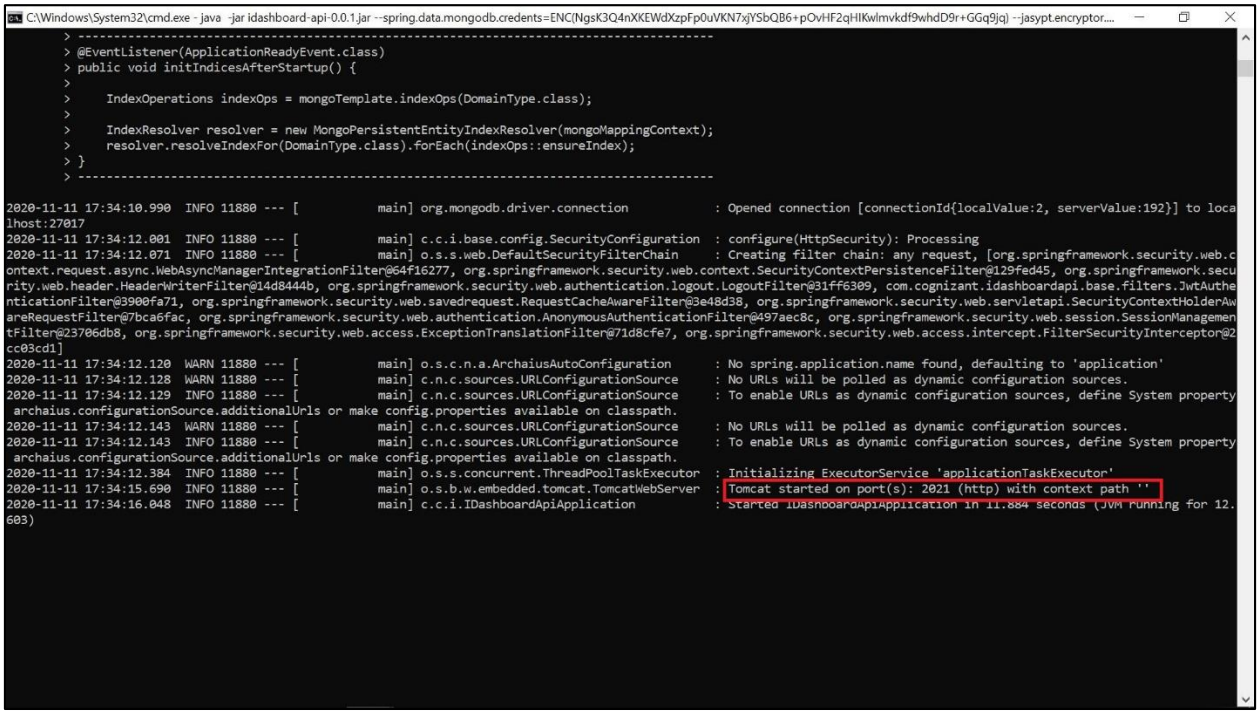

Figure 15: Validate

1.1.7. Execute auth-api-3.0.0.jar

1. Open the command prompt and navigate to **bin** folder (Refer the section Binaries/Setup Files)

   java -jar auth-api-3.0.0.jar --spring.data.mongodb.credents=**ENC(ecrypted_key)** --jasypt.encryptor.password=**<encoded_jasypt_pass>**

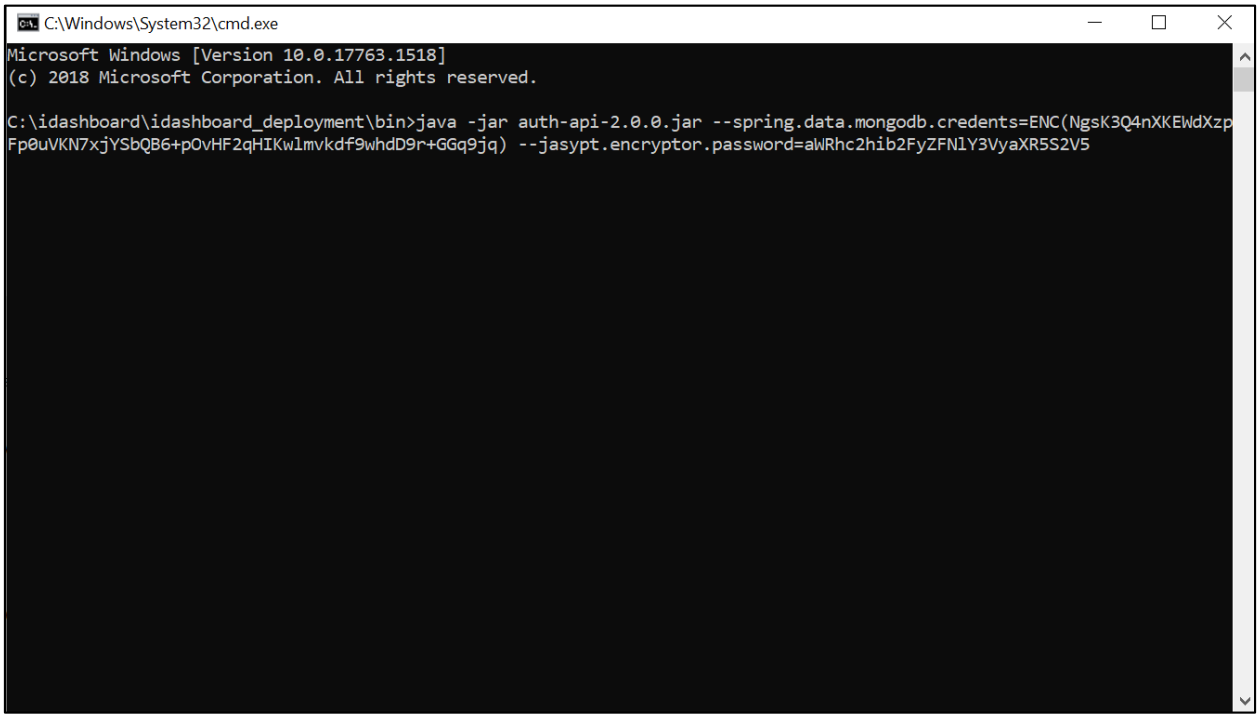

Figure 16: auth api jar

2. Validate the port that **auth-api** is listening (Refer the section NGINX Configuration, to validate the port in **nginx.conf** file)

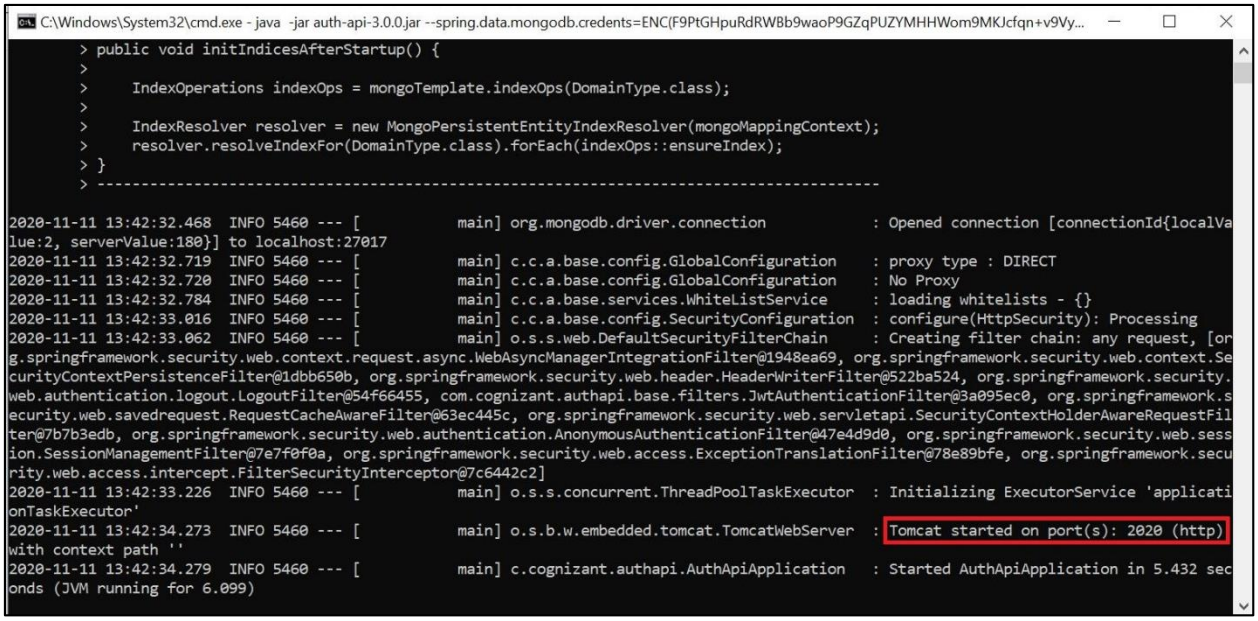

Figure 17: Validate

- To change the port, use **–server.port=<port-number>**
- To change the db name, use **-- spring.data.mongodb.uri= mongodb://<username>:${spring.data.mongodb.credents}@localhost/<database-name>**

1.1.8. Execute Collectors to extract data for data sources

1. Encrypt the password/tokens using **jasypt** library. Refer the section Encrypt Password and use the below commands to run the respective collectors
2. Open the command and navigate to respective collector jar files (Refer the section Binaries/Setup Files)

Jenkins

- java -jar ciqdashboard-data-collector-jenkins-0.0.1.jar --spring.data.mongodb.credents=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5 --jenkins.url=<Jenkins-url> --jenkins.username=<username> --jenkins.token= ENC (<encrypt-token>)

GITHUB

- java -jar ciqdashboard-data-collector-github-0.0.1.jar --spring.data.mongodb.credents=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlm

vkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5 --github.url=https://api.github.com --github.token= ENC(<encrypt-token>)

GITLAB

- java -jar ciqdashboard-data-collector-gitlab-0.0.1.jar --spring.data.mongodb.credents=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5 --gitlab.url=<gitlab-url> --gitlab.token= ENC(<encrypt-token>)

Artifactory

- java -jar ciqdashboard-data-collector-artifactory-0.0.1.jar --spring.data.mongodb.credents=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5 --artifactory.url=<atifactory-url> --artifactory.token== ENC(<encrypt-token>) --artifactory.username=<username>

JIRA

- java -jar ciqdashboard-data-collector-jira-0.0.1.jar --spring.data.mongodb.credents=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5 --jiraServer.url=http://<jiraServer-url>/rest/api/latest --jiraServer.username=<username> --jiraServer.password=ENC(<encrypt-password>)

Microfocus ALM

- java -jar ciqdashboard-data-collector-alm-0.0.1.jar --spring.data.mongodb.credents=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5 --almServer.url=http://<alm-url>/qcbin --almServer.username=<username> --almServer.password= ENC(<encrypt-password>)

SONARQUBE

- java -jar ciqdashboard-data-collector-sonarqube-0.0.1.jar --spring.data.mongodb.credents=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5 --sonarqube.url=<sonarcube-url> --sonarqube.username=<username> --sonarqube.token= ENC(<encrypt-token>)

```
        >
        >     IndexOperations indexOps = mongoTemplate.indexOps(DomainType.class);
        >
        >     IndexResolver resolver = new MongoPersistentEntityIndexResolver(mongoMappingContext);
        >     resolver.resolveIndexFor(DomainType.class).forEach(indexOps::ensureIndex);
        > }
        > ---------------------------------------------------------------------------------------

11-11-2020 13:28:13.985 [35m[main][0;39m [34mINFO [0;39m org.mongodb.driver.connection.info - Opened connection [con
nectionId{localValue:2, serverValue:174}] to localhost:27017
11-11-2020 13:28:14.157 [35m[main][0;39m [31mWARN [0;39m com.netflix.config.sources.URLConfigurationSource.<init> -
No URLs will be polled as dynamic configuration sources.
11-11-2020 13:28:14.157 [35m[main][0;39m [34mINFO [0;39m com.netflix.config.sources.URLConfigurationSource.<init> -
To enable URLs as dynamic configuration sources, define System property archaius.configurationSource.additionalUrls or m
ake config.properties available on classpath.
11-11-2020 13:28:14.169 [35m[main][0;39m [31mWARN [0;39m com.netflix.config.sources.URLConfigurationSource.<init> -
No URLs will be polled as dynamic configuration sources.
11-11-2020 13:28:14.170 [35m[main][0;39m [34mINFO [0;39m com.netflix.config.sources.URLConfigurationSource.<init> -
To enable URLs as dynamic configuration sources, define System property archaius.configurationSource.additionalUrls or m
ake config.properties available on classpath.
11-11-2020 13:28:14.317 [35m[main][0;39m [34mINFO [0;39m org.springframework.scheduling.concurrent.ThreadPoolTaskSch
eduler.initialize - Initializing ExecutorService 'taskScheduler'
11-11-2020 13:28:14.554 [35m[main][0;39m [34mINFO [0;39m com.cognizant.dashboard.collectors.jenkins.JenkinsCollector
Application.logStarted - Started JenkinsCollectorApplication in 3.676 seconds (JVM running for 4.218)
11-11-2020 13:29:00.002 [35m[scheduling-1][0;39m [34mINFO [0;39m com.cognizant.dashboard.collectors.jenkins.schedule
r.JobScheduler.cronScheduler - ************************************cron <Start>*********************************
*********************************
11-11-2020 13:29:00.004 [35m[scheduling-1][0;39m [34mINFO [0;39m com.cognizant.dashboard.collectors.jenkins.schedule
r.JobSchedulerImpl.beforeJob - Before Job process........!
```

Figure 18: Collectors

3. Run all the collectors and wait for the scheduler to complete the job.
4. After running the collectors, refresh the database and verify the collectors.

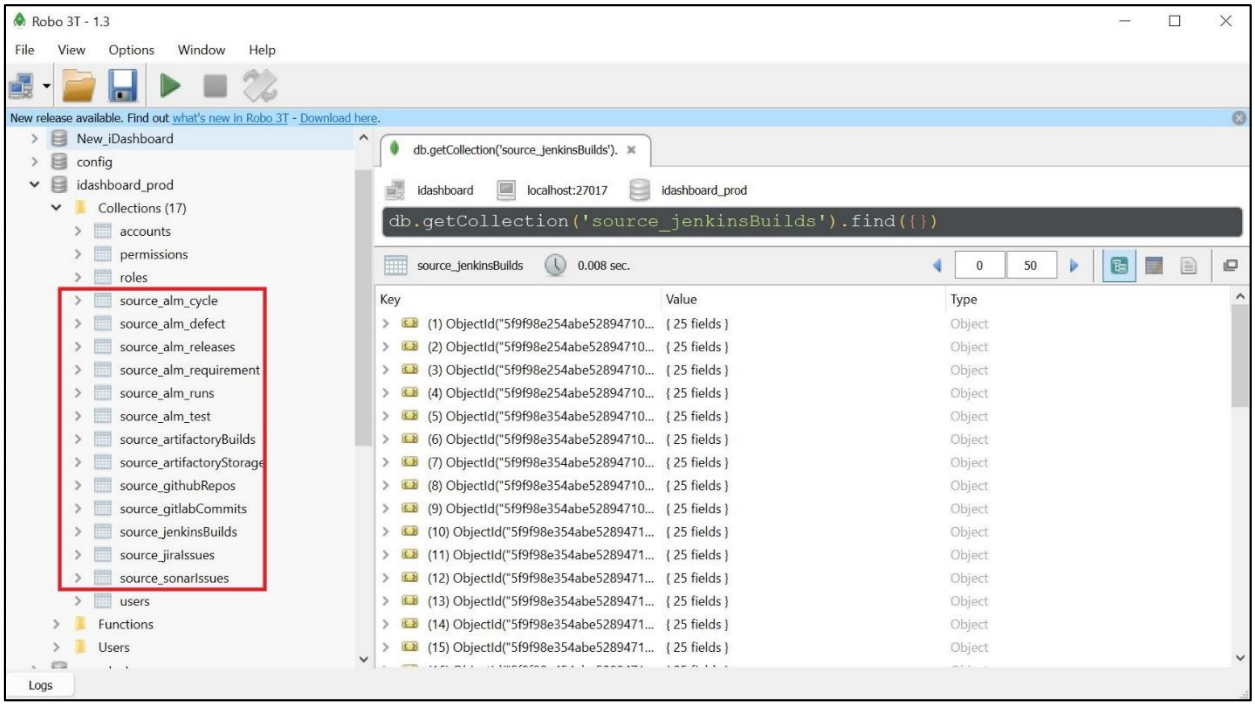

Figure 19: Refresh and verify

Note

Jasypt.encryptor.password as in **step no: 6** of the section Encrypt Password

1.1.9. NGINX Configuration

Follow the steps below after downloading the software. Refer the section Software Requirements for downloading NGINX.

1. Extract to any folder (**Example: C:\ Drive**)
2. From **binaries**(Refer the section Binaries/Setup Files)**->nginx**, open c**iqdashboard.conf** file, copy the entire available content and paste it in **nginx_folder->conf->nginx.conf**

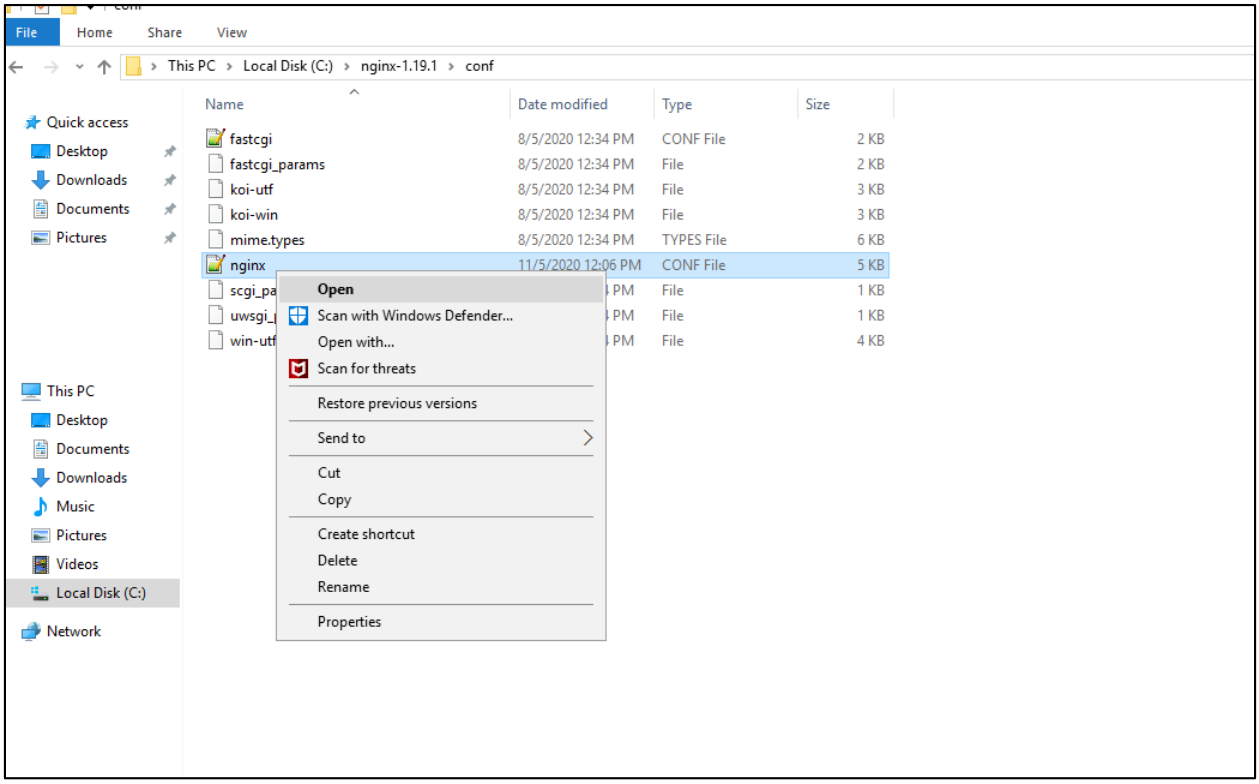

Figure 20: ngnix.conf

Edit the UI binaries path in **nginx.conf** file and change the **listen** port to **2022**

```
    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;


server {
    listen       2022;
    server_name  localhost;
    charset utf-8;
    access_log off;
    server_tokens off;
    client_max_body_size 100M;
    error_page  400  /ui/index.html;
    set $Headers 'Authorization,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';
    location /ui {
        root C:\\idashboard\\idashboard_deployment\\binaries;
        try_files $uri $uri/ /index.html;

        add_header Content-Security-Policy "default-src 'self' 'unsafe-eval';script-src-elem 'self' ;
        style-src 'self' 'unsafe-inline' https://fonts.googleapis.com;font-src https://fonts.gstatic.com 'self' data:;img-src 'self' data: blob:; base-uri 'self';
        add_header X-Frame-Options SAMEORIGIN;
        add_header X-XSS-Protection "1; mode=block";
        add_header X-Content-Type-Options "nosniff" always;
        add_header Referrer-Policy "strict-origin" always;
        add_header Feature-Policy "microphone 'none'; geolocation 'none'; camera 'none'" always;
      }
    location /api/idashboard {

        proxy_pass http://localhost:2021/;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Forwarded-Host $server_name;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Connection '';
```

Figure 21: listen port

Open the command prompt as Administrator. Navigate to **nginx folder** path extracted in **C: drive** and use command: **start nginx** to start the UI.

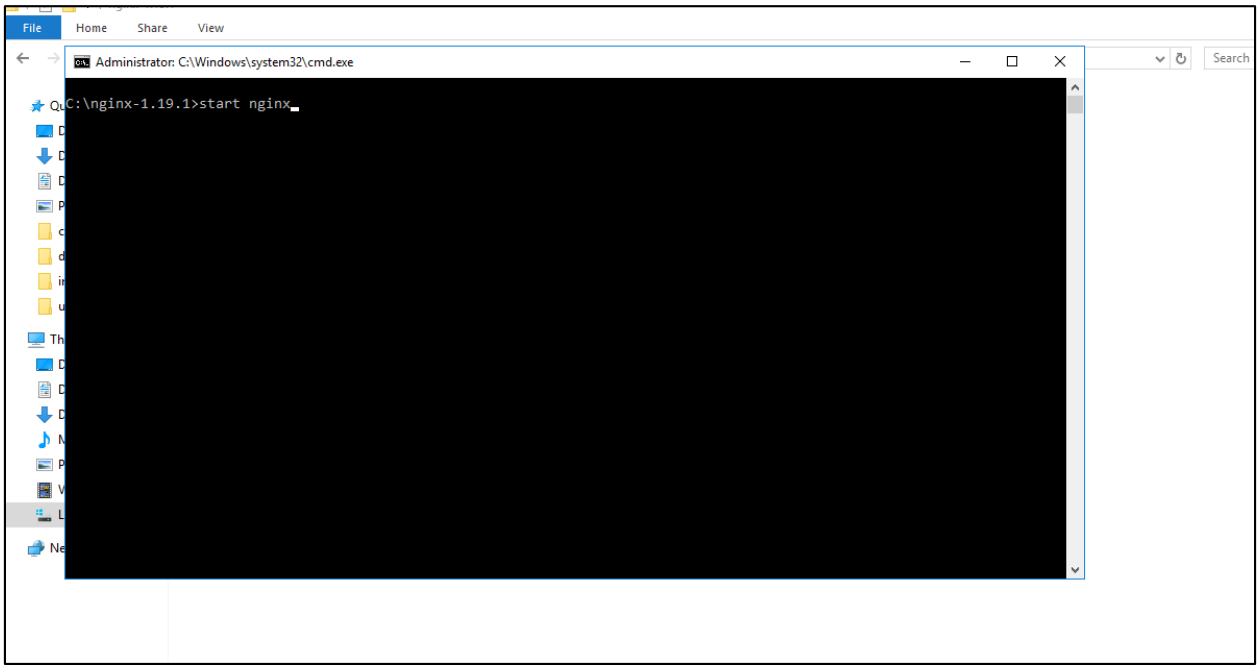

Figure 22: start nginx

Open http://localhost:2022/ui to access CIQDashboard. Replace localhost with IP, to access from different machine/network

1.1.10. Apache server Configuration

Follow the steps below after downloading the software. Refer the section Software Requirements for downloading Apache Server

1. Extract to any folder (**Example: C:\ Drive**)
2. From **binaries**(Refer the section Binaries/Setup Files)**->apache**, open c**iqdashboard.conf** file, copy the entire available content and paste it in **apache_folder->conf->httpd.conf**

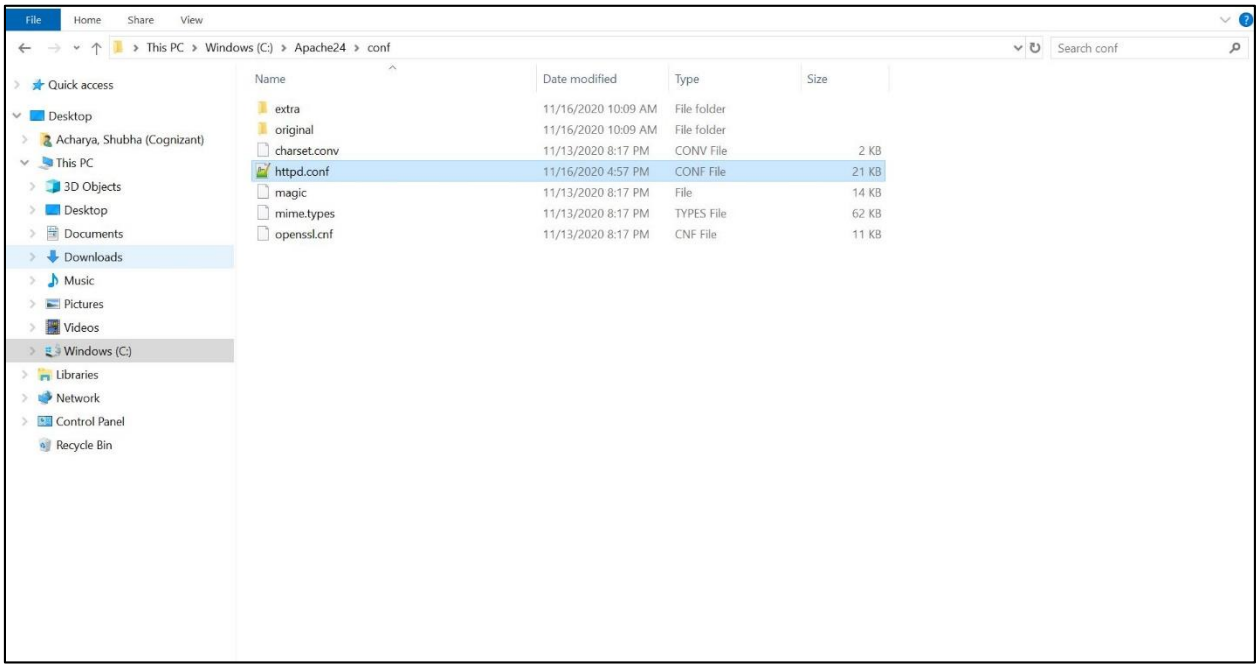

Figure 23: httpd.conf

3. Open a command prompt window and cd to the c:\Apache24\bin folder.
4. To Start Apache in the command prompt type:

   >httpd.exe
5. You can test your installation by opening up your Browser and typing in the address:

   http://localhost
6. You can shut down Apache by pressing Ctrl+C (It may take a few seconds)
7. To install as a service. Open command prompt as Administrator and type:

   >httpd.exe -k install
8. You can start/stop the service with the command:

   >services.msc

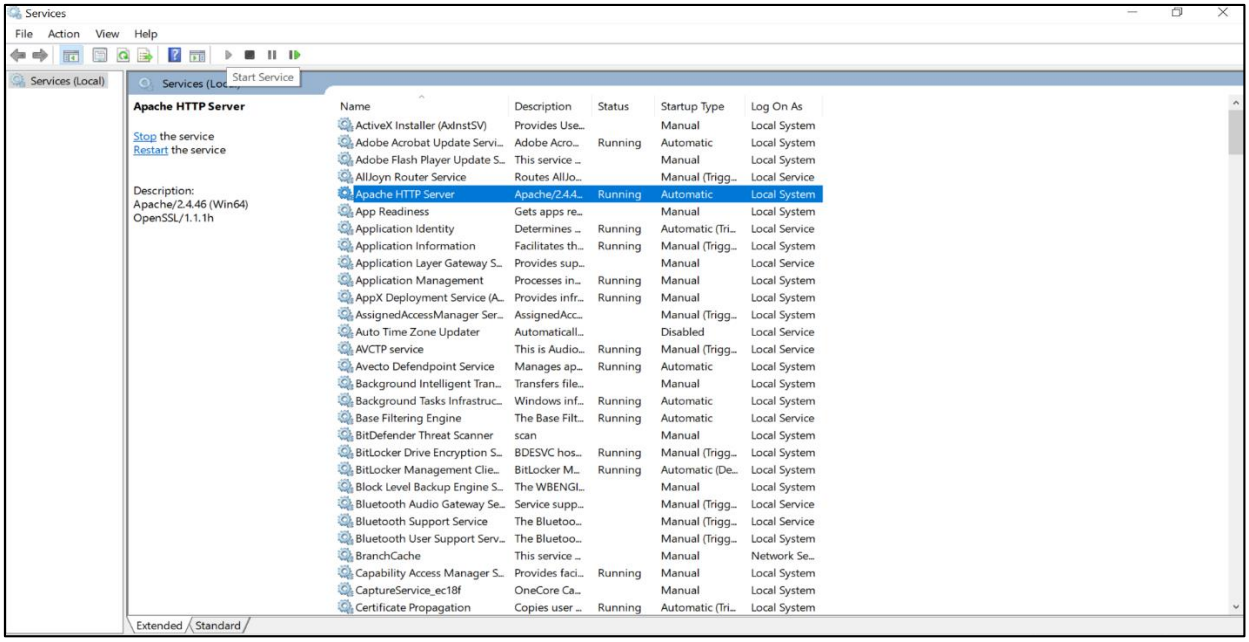

Figure 24: services.msc

9. Place the UI binaries in to C:\Apache24\htdocs folder

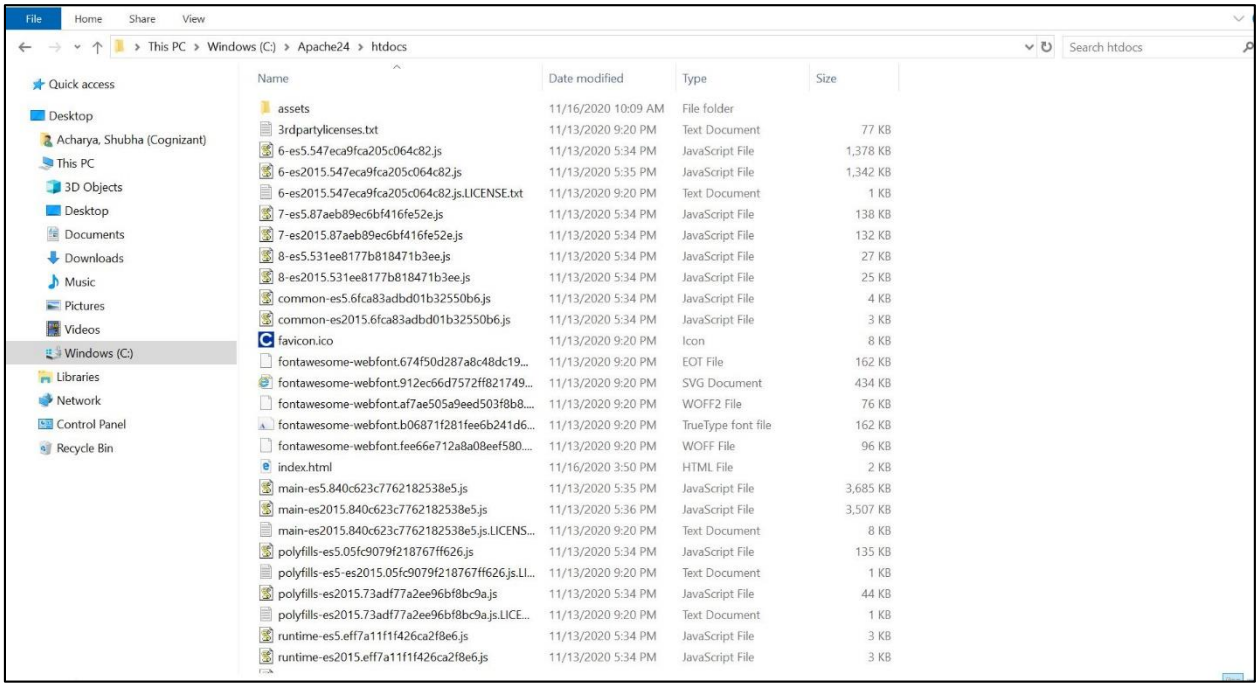

Figure 25: htdocs

10. Change the port number in **httpd.conf** if there is any change in port number.

```
510
511 # Virtual hosts
512 #Include conf/extra/httpd-vhosts.conf
513
514 # Local access to the Apache HTTP Server Manual
515 #Include conf/extra/httpd-manual.conf
516
517 # Distributed authoring and versioning (WebDAV)
518 #Include conf/extra/httpd-dav.conf
519
520 # Various default settings
521 #Include conf/extra/httpd-default.conf
522 # Configure mod_proxy_html to understand HTML4/XHTML1
523 <IfModule proxy_html_module>
524 Include conf/extra/proxy-html.conf
525 </IfModule>
526
527 # Secure (SSL/TLS) connections
528 #Include conf/extra/httpd-ssl.conf
529 #
530 # Note: The following must must be present to support
531 #       starting without SSL on platforms with no /dev/random equivalent
532 #       but a statically compiled-in mod_ssl.
533 #
534 <IfModule ssl_module>
535 SSLRandomSeed startup builtin
536 SSLRandomSeed connect builtin
537 </IfModule>
538
539 <VirtualHost *:80>
540 # Enable Proxy
541 ProxyPreserveHost On
542 # Incoming. Example host and port, http://PAS_SERVER:8080/
543 # Only enable below what you like to be accessed from outside
544 ProxyPassMatch "/api/idashboard(.*)$" "http://localhost:2021/$1"
545 ProxyPassReverse "/api/idashboard(.*)$" "http://localhost:2021/$1"
546 ProxyPassMatch "/api(.*)$" "http://localhost:2020/$1"
547 ProxyPassReverse "/api(.*)$" "http://localhost:2020/$1"
548 </VirtualHost>
```

Figure 26: httpd.config

11. Start the server. Open http://localhost to access CIQDashboard. Replace localhost with IP, to access from different machine/network

FAQs

1. How to change the default port?

   To change the default port, use **--server.port=<port-number>**. (Refer section 6.2.2 and 6.2.3)

2. How can I change the database name?

   To change the db name, use **-- spring.data.mongodb.uri mongodb://<username>:${spring.data.mongodb.credents}@localhost/<database-name> =** (Refer section 6.2.2 , 6.2.3 and 6.2.4)

3. How to schedule the collectors?

   By default, the scheduler runs for every minute. To change the scheduler time, **use --scheduler.cron = <time in cron expression>**. To know more about cron expression, refer https://docs.oracle.com/cd/E12058_01/doc/doc.1014/e12030/cron_expressions.htm