



Integrated Quality Dashboard

INSTALLATION GUIDE

WINDOWS OS

Version 3.1

Contents

1.	About this Installation Guide	4
2.	About Cognizant® Integrated Quality Dashboard.....	5
2.1.	Product Overview	5
3.	Hardware Requirements.....	6
4.	Software Requirements	7
5.	Binaries/Setup Files.....	8
6.	Installation Procedure in Windows OS	9
6.1.	Mongo DB Configuration	9
6.1.1.	Installing Mongo DB	9
6.1.2.	Authenticating Mongo DB.....	9
6.1.3.	Validating Admin DB in RoboMongo	12
6.1.4.	Configuring MongoDB for CIQDashboard.....	15
6.2.	Configuration of CIQDashboard.....	15
6.2.1.	Encrypt Password.....	15
6.2.2.	Execute ciqdashboard-api-0.0.1.jar	17
6.2.3.	Execute auth-api-3.0.0.jar.....	19
6.2.4.	Execute Collectors to extract data for data sources	20
6.2.5.	NGINX Configuration.....	23
6.2.6.	Apache server Configuration	25
7.	FAQs	29

Legal Notice

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® Integrated Quality Dashboard 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.

1. About this Installation Guide

The Cognizant® Integrated Quality Dashboard (CIQDashboard) Installation Guide, provides help to install CIQDashboard Server in your system. It guides you through the steps to install and uninstall the CIQDashboard 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® Integrated Quality Dashboard	Provides information about CIQDashboard
Hardware Requirements	Provides an overview about required hardware configuration
Software Requirements	Provides an overview about prerequisites for installing CIQDashboard
Binaries/Setup Files	Provides information on the required binaries/setup files
Installation Procedure in Windows OS	Provides procedure to install CIQDashboard
Uninstallation Procedure	Provides procedure to uninstall CIQDashboard
FAQs	Provides answers to a list of commonly asked questions regarding CIQDashboard

2. About Cognizant® Integrated Quality Dashboard

This chapter consists:

- [Product Overview](#)

2.1. Product Overview

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

3. 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

4. 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
--	---

5. Binaries/Setup Files

- Please build the binaries from the respective repositories.
- 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

6. Installation Procedure in Windows OS

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

6.1. Mongo DB Configuration

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

6.1.1. Installing Mongo DB

Follow the steps below after downloading the software. Refer the section [Software Prerequisites](#) for downloading Mongo DB.

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

6.1.2. Authenticating Mongo DB

Follow the steps below to authenticate Mongo DB server.

6.1.2.1. Start Mongo Server in Normal Mode:

- Create a folder with name: **mongoDB_Data**** in **C:**.
- 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.
- In the command prompt, type **mongod --dbpath c:\mongoDB_Data** and press **Enter**. The Mongo server starts in normal mode.

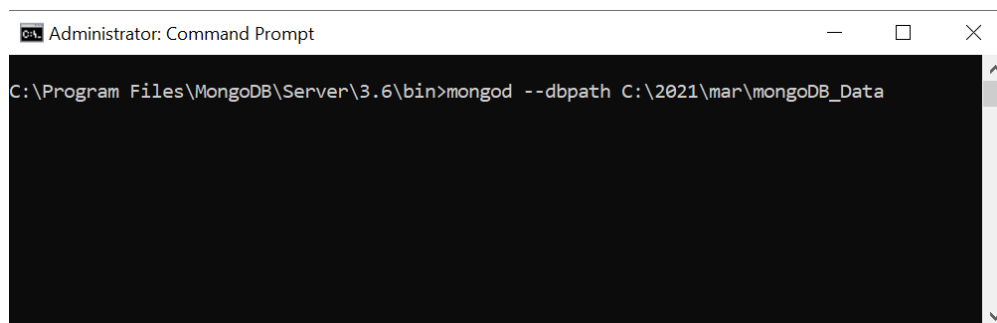


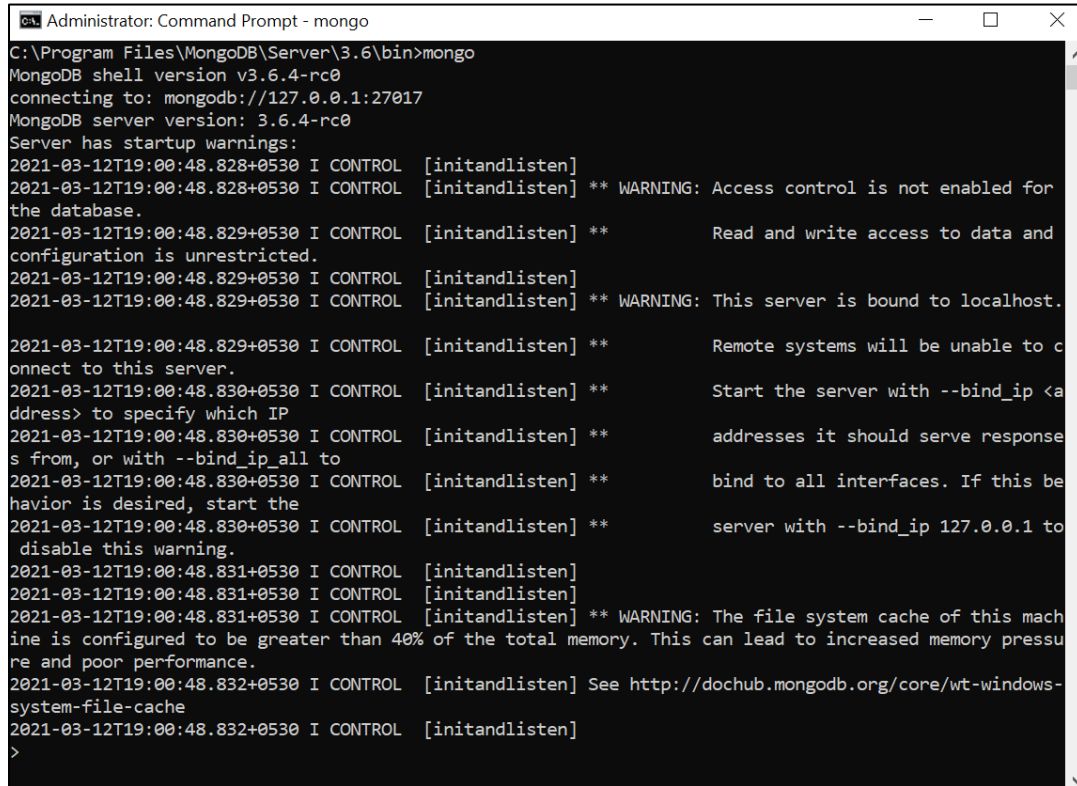
Figure 1: Administrator Command Prompt



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

6.1.2.2. 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.



```
Administrator: Command Prompt - mongo
C:\Program Files\MongoDB\Server\3.6\bin>mongo
MongoDB shell version v3.6.4-rc0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.4-rc0
Server has startup warnings:
2021-03-12T19:00:48.828+0530 I CONTROL [initandlisten]
2021-03-12T19:00:48.828+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for
the database.
2021-03-12T19:00:48.829+0530 I CONTROL [initandlisten] **      Read and write access to data and
configuration is unrestricted.
2021-03-12T19:00:48.829+0530 I CONTROL [initandlisten]
2021-03-12T19:00:48.829+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
Remote systems will be unable to connect to this server.
2021-03-12T19:00:48.830+0530 I CONTROL [initandlisten] **      Start the server with --bind_ip <address> to specify which IP
addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the
server with --bind_ip 127.0.0.1 to disable this warning.
2021-03-12T19:00:48.831+0530 I CONTROL [initandlisten]
2021-03-12T19:00:48.831+0530 I CONTROL [initandlisten]
2021-03-12T19:00:48.831+0530 I CONTROL [initandlisten] ** WARNING: The file system cache of this machine is configured to be greater than 40% of the total memory. This can lead to increased memory pressure and poor performance.
2021-03-12T19:00:48.832+0530 I CONTROL [initandlisten] See http://dochub.mongodb.org/core/wt-windows-system-file-cache
2021-03-12T19:00:48.832+0530 I CONTROL [initandlisten]
>
```

Figure 2: Mongo

6.1.2.3. Creating Admin User and Password:

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



```
Administrator: Command Prompt - mongo
> use admin
switched to db admin
>
```

Figure 3: use admin

- 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" }]})



```
Administrator: Command Prompt - mongo
> use admin
switched to db admin
> db.createUser({ user: "admin", pwd: "adminpassword", roles: [{ role: "root", db: "admin"
" db.createUser({ user: "admin", pwd: "adminpassword", roles: [{ role: "root", db: "admin
" }])
Successfully added user: {
  "user" : "admin",
  "roles" : [
    {
      "role" : "root",
      "db" : "admin"
    }
  ]
}
```

Figure 4: Admin user

- 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.

6.1.2.4. Create Custom DB and its Users:

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

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



```
Administrator: Command Prompt - mongo
@(<shell>):1:1
> db.createUser(
...  {
...    user: "ciqdashboard",
...    pwd: "ciqdashboard",
...    roles:
...    [
...      { role: "readWrite", db: "ciqdashboard_prod" }
...    ]
...  }
... )
Successfully added user: {
  "user" : "ciqdashboard",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "ciqdashboard_prod"
    }
  ]
}
```

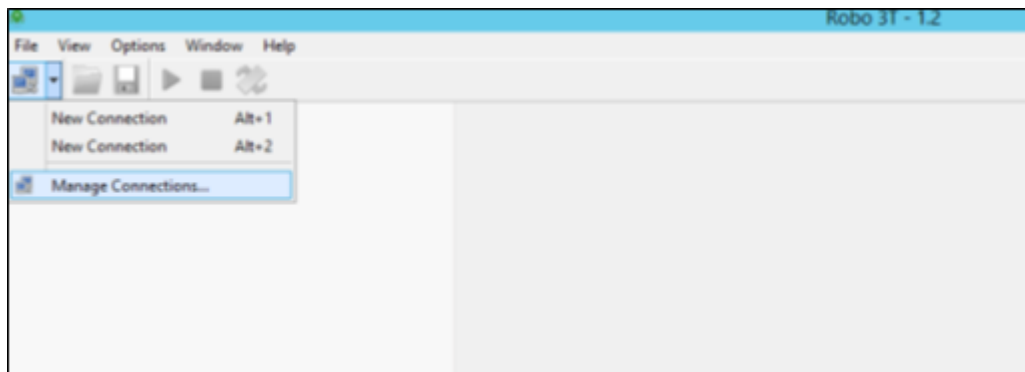
Figure 5: ciqdashboard_prod

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

6.1.3. Validating Admin DB in RoboMongo

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

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

*Figure 6: Manage connections*

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

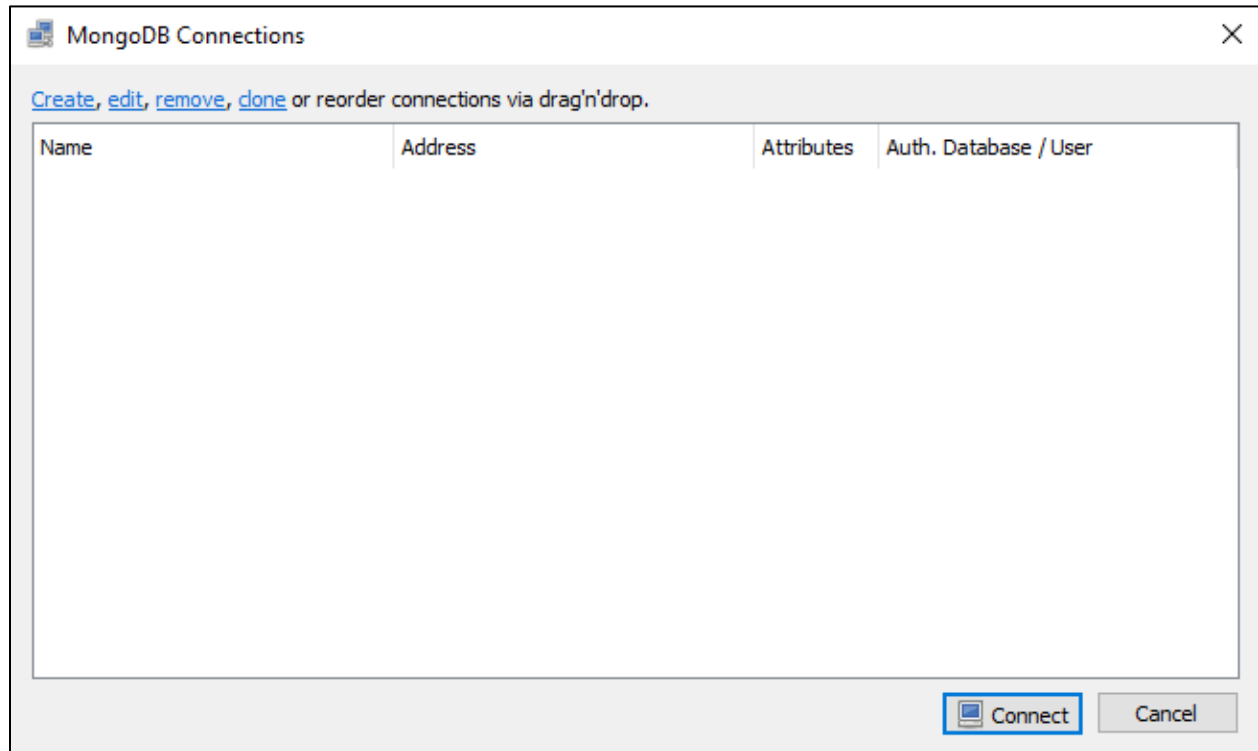


Figure 7: Create

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

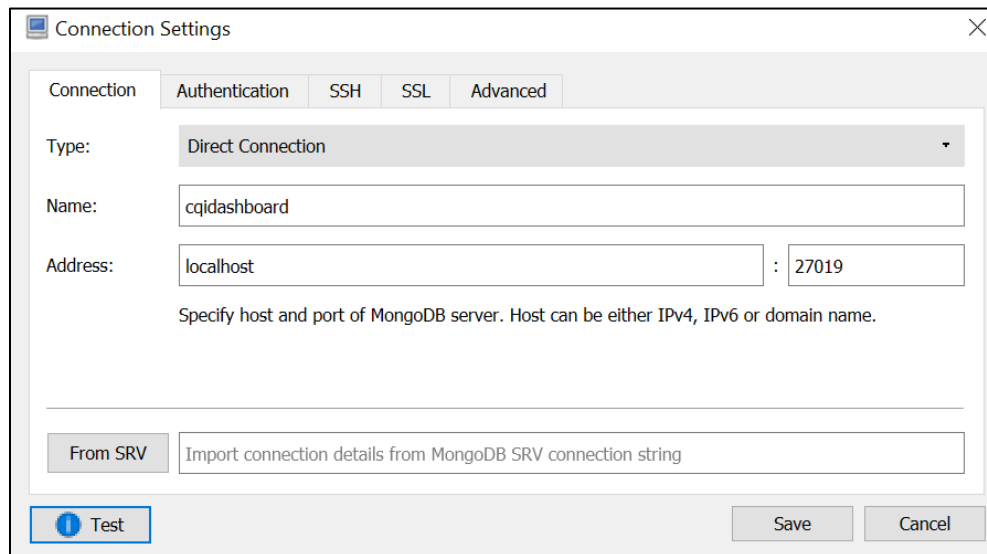
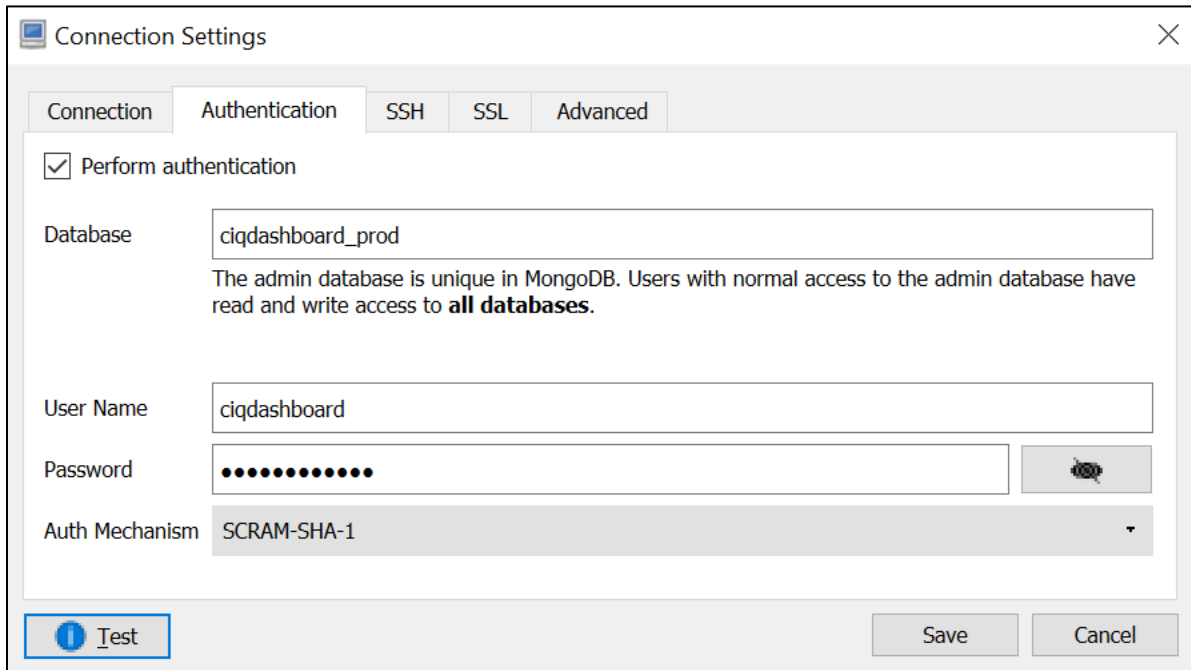


Figure 8: Connection

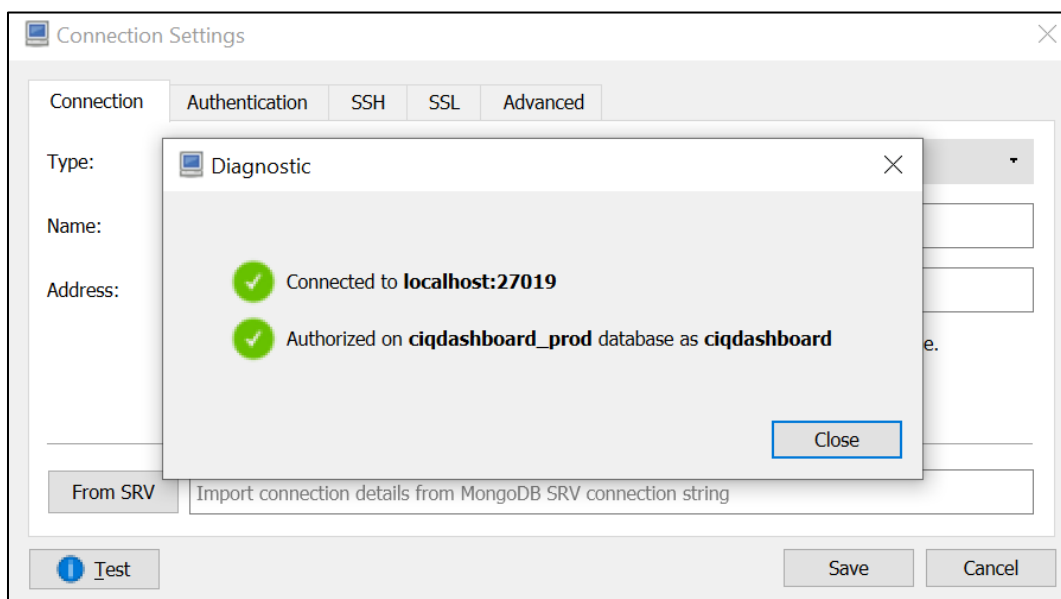
- In **Name**, type a name for the connection you are creating.
- In **Address**, type the host address and the port.
- Click **Authentication** tab. The **Authentication** pane appears.



The screenshot shows the 'Connection Settings' dialog box with the 'Authentication' tab selected. The 'Perform authentication' checkbox is checked. The 'Database' field contains 'ciqdashboard_prod' with a note below it: 'The admin database is unique in MongoDB. Users with normal access to the admin database have read and write access to **all databases**.' The 'User Name' field contains 'ciqdashboard'. The 'Password' field is masked with dots. The 'Auth Mechanism' dropdown is set to 'SCRAM-SHA-1'. At the bottom, there are 'Test', 'Save', and 'Cancel' buttons.

Figure 9: Authentication

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



The screenshot shows the 'Connection Settings' dialog box with the 'Authentication' tab selected. A 'Diagnostic' pop-up window is displayed in the foreground. The pop-up shows two green checkmarks: 'Connected to localhost:27019' and 'Authorized on ciqdashboard_prod database as ciqdashboard'. A 'Close' button is at the bottom right of the pop-up. In the background, the 'Connection Settings' dialog box shows the 'Test' button highlighted.

Figure 10: Diagnostics

6.1.4. Configuring MongoDB for CIQDashboard

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



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

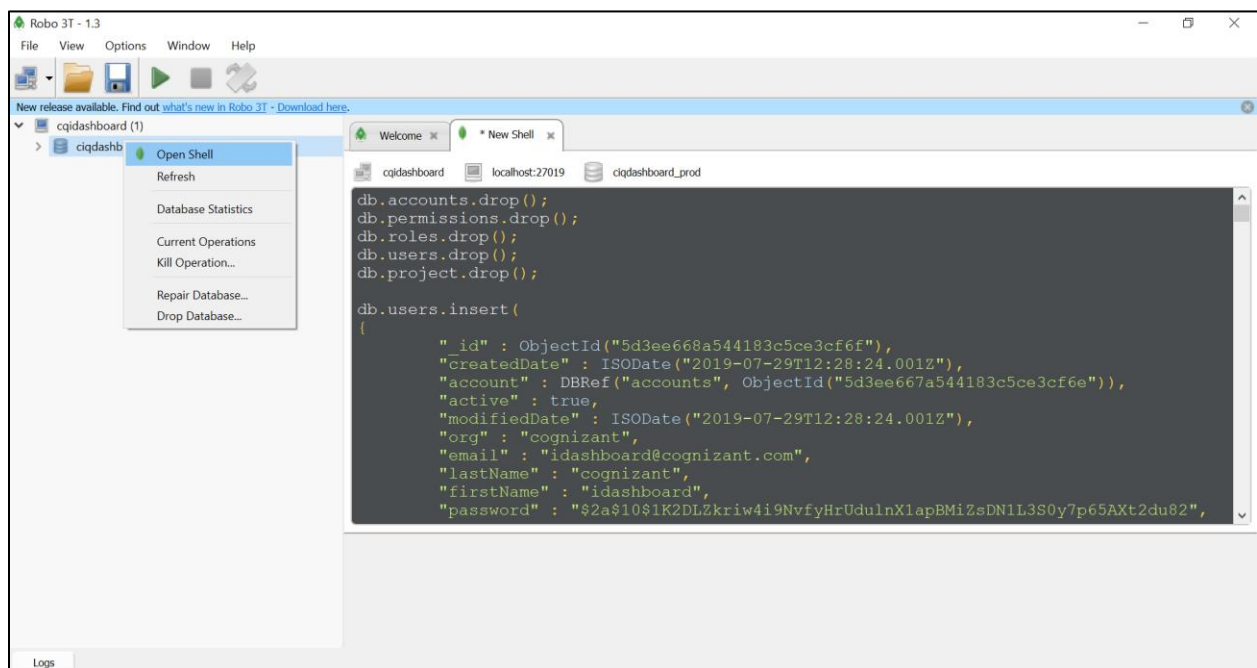


Figure 11: Execute



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

6.2. Configuration of CIQDashboard

This section describes the steps to configure CIQDashboard.

6.2.1. Encrypt Password

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

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

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



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

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17763.1518]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\idashboard\idashboard_deployment\jasypt\jasypt-1.9.3-dist\jasypt-1.9.3>java -cp lib/jasypt-1.9.3.jar org.jasypt.intf.
cli.JasyptPBEStrEncryptionCLI password=idashboardSecurityKey algorithm=PBEWITHHMACSHA512ANDAES_256 input=idashboard i
vGeneratorClassName=org.jasypt.iv.RandomIvGenerator

---ENVIRONMENT-----
Runtime: Oracle Corporation Java HotSpot(TM) 64-Bit Server VM 11.0.9+7-LTS

---ARGUMENTS-----
input: idashboard
password: idashboardSecurityKey
ivGeneratorClassName: org.jasypt.iv.RandomIvGenerator
algorithm: PBEWITHHMACSHA512ANDAES_256

---OUTPUT-----
NgsK3Q4nXKEwdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKw1mvkdf9whdD9r+GGq9jq

C:\idashboard\idashboard_deployment\jasypt\jasypt-1.9.3-dist\jasypt-1.9.3>
```

Figure 12: Output

- The output is the encrypted key, copy and save the key.
- Encode the security key (ciqdashboardSecurityKey - aWRhc2hib2FyZFNIY3VyaXR5S2V5) with <https://www.base64encode.org/> and pass it in the command line as --
jasypt.encryptor.password=<encoded_jasypt_pass>

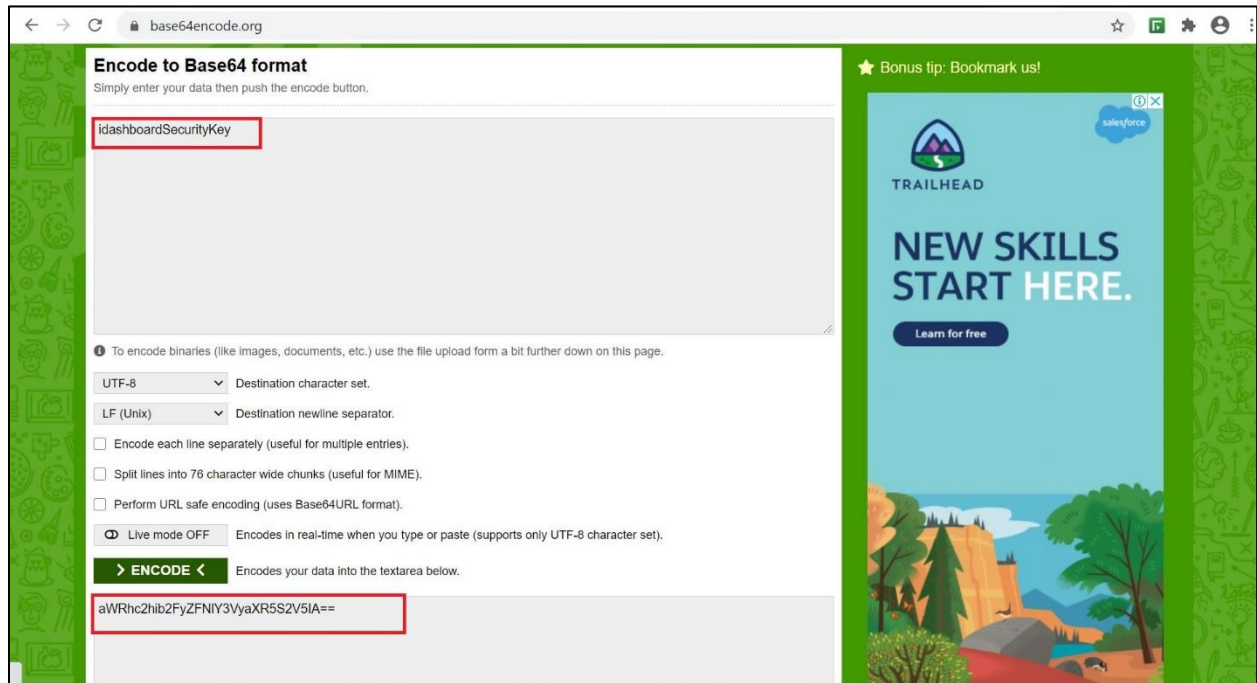


Figure 13: Encode

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

6.2.2. Execute ciqdashboard-api-0.0.1.jar

- 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.credentials=ENC(<encrypted_key>) --jasypt.encryptor.password=<encoded_jasypt_pass>

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17763.1518]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\idashboard\idashboard_deployment\bin>java -jar idashboard-api-2.0.0.jar --spring.data.mongodb.credentials=ENC(NgsK3Q4nXK
EwdXzpFp0uVKN7xjYsBQ86+pOvHF2qHlKwlmvkdF9whdD9r+GGq9jq) --jasypt.encryptor.password=awRhC2hib2FyZFNLy3VyaXR5S2V5

```

Figure 14: cidashboard api jar

- Validate the port that **cidashboard-api** is listening (Refer the section [NGINX Configuration](#), to validate the port in **nginx.conf** file)

```

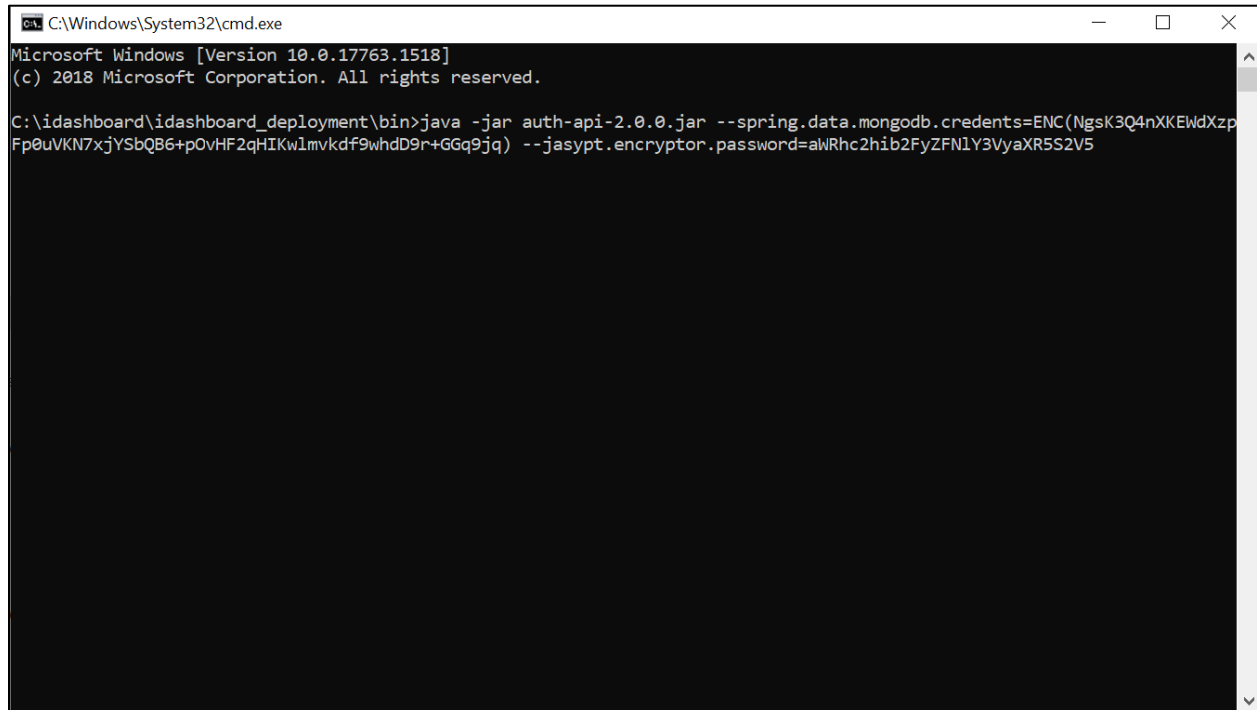
C:\Windows\System32\cmd.exe - java -jar idashboard-api-0.0.1.jar --spring.data.mongodb.credentials=ENC(NgsK3Q4nXKWEwdXzpFp0uVKN7xjYsBQ86+pOvHF2qHlKwlmvkdF9whdD9r+GGq9jq) --jasypt.encryptor...
> @EventListener(ApplicationReadyEvent.class)
> public void initIndicesAfterStartup() {
>
>     IndexOperations indexOps = mongoTemplate.indexOps(DomainType.class);
>
>     IndexResolver resolver = new MongoPersistentEntityIndexResolver(mongoMappingContext);
>     resolver.resolveIndexFor(DomainType.class).forEach(indexOps::ensureIndex);
> }
>
-----
2020-11-11 17:34:10.990 INFO 11880 --- [main] org.mongodb.driver.connection : Opened connection [connectionId{localValue:2, serverValue:192}] to localhost:27017
2020-11-11 17:34:12.001 INFO 11880 --- [main] c.c.i.base.config.SecurityConfiguration : configure(HttpSecurity): Processing
2020-11-11 17:34:12.071 INFO 11880 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter@64f16277, org.springframework.security.web.context.SecurityContextPersistenceFilter@129fed45, org.springframework.security.web.header.HeaderWriterFilter@14d8444b, org.springframework.security.web.authentication.logout.LogoutFilter@31ff6309, com.cognizant.idashboardapi.base.filters.JwtAuthenticationFilter@3900fa71, org.springframework.security.web.savedrequest.RequestCacheAwareFilter@3e48d38, org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter@7bca6fac, org.springframework.security.web.authentication.AnonymousAuthenticationFilter@497aec8c, org.springframework.security.web.session.SessionManagementFilter@23706db8, org.springframework.security.web.access.ExceptionTranslationFilter@71d8cfe7, org.springframework.security.web.access.intercept.FilterSecurityInterceptor@2cc03cd1]
2020-11-11 17:34:12.120 WARN 11880 --- [main] o.s.c.n.a.ArchaiusAutoConfiguration : No spring.application.name found, defaulting to 'application'
2020-11-11 17:34:12.128 WARN 11880 --- [main] c.n.c.sources.URLConfigurationSource : No URLs will be polled as dynamic configuration sources.
2020-11-11 17:34:12.129 INFO 11880 --- [main] c.n.c.sources.URLConfigurationSource : To enable URLs as dynamic configuration sources, define System property archaius.configurationSource.additionalUrls or make config.properties available on classpath.
2020-11-11 17:34:12.143 WARN 11880 --- [main] c.n.c.sources.URLConfigurationSource : No URLs will be polled as dynamic configuration sources.
2020-11-11 17:34:12.143 INFO 11880 --- [main] c.n.c.sources.URLConfigurationSource : To enable URLs as dynamic configuration sources, define System property archaius.configurationSource.additionalUrls or make config.properties available on classpath.
2020-11-11 17:34:12.384 INFO 11880 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-11-11 17:34:15.690 INFO 11880 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 2021 (http) with context path ''
2020-11-11 17:34:16.048 INFO 11880 --- [main] c.c.i.IDashboardApiApplication : Started IDashboardApiApplication in 11.884 seconds (JVM running for 12.693)

```

Figure 15: Validate

6.2.3. Execute auth-api-3.0.0.jar

- 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.credentials=ENC(encrypted_key) --jasypt.encryptor.password=<encoded_jasypt_pass>`



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17763.1518]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\idashboard\idashboard_deployment\bin>java -jar auth-api-2.0.0.jar --spring.data.mongodb.credentials=ENC(NgsK3Q4nXKEWdXzp
Fp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdF9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNlY3VyaXR5S2V5
```

Figure 16: auth api jar


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

```

C:\Windows\System32\cmd.exe - java -jar auth-api-3.0.0.jar --spring.data.mongodb.credentials=ENC(F9PtGHpuRdRW8b9waoP9GZqPUZYMHWHom9MKJcfqn+v9Vy...
> public void initIndicesAfterStartup() {
>
>     IndexOperations indexOps = mongoTemplate.indexOps(DomainType.class);
>
>     IndexResolver resolver = new MongoPersistentEntityIndexResolver(mongoMappingContext);
>     resolver.resolveIndexFor(DomainType.class).forEach(indexOps::ensureIndex);
> }
> -----
2020-11-11 13:42:32.468 INFO 5460 --- [main] org.mongodb.driver.connection : Opened connection [connectionId{localVa
lue:2, serverValue:180}] to localhost:27017
2020-11-11 13:42:32.719 INFO 5460 --- [main] c.c.a.base.config.GlobalConfiguration : proxy type : DIRECT
2020-11-11 13:42:32.720 INFO 5460 --- [main] c.c.a.base.config.GlobalConfiguration : No Proxy
2020-11-11 13:42:32.784 INFO 5460 --- [main] c.c.a.base.services.WhiteListService : loading whitelists - {}
2020-11-11 13:42:33.016 INFO 5460 --- [main] c.c.a.base.config.SecurityConfiguration : configure(HttpSecurity): Processing
2020-11-11 13:42:33.062 INFO 5460 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [or
g.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter@1948ea69, org.springframework.security.web.context.Se
curityContextPersistenceFilter@1dbb650b, org.springframework.security.web.header.HeaderWriterFilter@522ba524, org.springframework.security.
web.authentication.logout.LogoutFilter@54f66455, com.cognizant.authapi.base.filters.JwtAuthenticationFilter@3a095ec0, org.springframework.s
ecurity.web.savedrequest.RequestCacheAwareFilter@63ec445c, org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFil
ter@7b7b3edb, org.springframework.security.web.authentication.AnonymousAuthenticationFilter@47e4d9d0, org.springframework.security.web.sess
ion.SessionManagementFilter@7e7f0f0a, org.springframework.security.web.access.ExceptionTranslationFilter@78e89bfe, org.springframework.secu
rity.web.access.intercept.FilterSecurityInterceptor@7c6442c2]
2020-11-11 13:42:33.226 INFO 5460 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicati
onTaskExecutor'
2020-11-11 13:42:34.273 INFO 5460 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 2020 (http)
with context path ''
2020-11-11 13:42:34.279 INFO 5460 --- [main] c.cognizant.authapi.AuthApiApplication : Started AuthApiApplication in 5.432 sec
onds (JVM running for 6.099)

```

Figure 17: Validate

	<ul style="list-style-type: none"> To change the port, use --server.port=<port-number> To change the db name, use --spring.data.mongodb.uri=mongodb://<username>:\${spring.data.mongodb.credentials}@localhost/<database-name>
---	--

6.2.4. Execute Collectors to extract data for data sources

- Encrypt the password/tokens using **jasypt** library. Refer the section [Encrypt Password](#) and use the below commands to run the respective collectors
- 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.credentials=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNIY3VyaXR5S2V5 --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.credentials=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNIY3VyaXR5S2V5 --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.credentials=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNIY3VyaXR5S2V5 --gitlab.url=<gitlab-url> --gitlab.token= ENC(<encrypt-token>)`

Artifactory

- `java -jar ciqdashboard-data-collector-artifactory-0.0.1.jar --spring.data.mongodb.credentials=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNIY3VyaXR5S2V5 --artifactory.url=<artifactory-url> --artifactory.token== ENC(<encrypt-token>) --artifactory.username=<username>`

JIRA

- `java -jar ciqdashboard-data-collector-jira-0.0.1.jar --spring.data.mongodb.credentials=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNIY3VyaXR5S2V5 --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.credentials=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNIY3VyaXR5S2V5 --almServer.url=http://<alm-url>/qcbn --almServer.username=<username> --almServer.password= ENC(<encrypt-password>)`

SONARQUBE

- `java -jar ciqdashboard-data-collector-sonarqube-0.0.1.jar --spring.data.mongodb.credentials=ENC(NgsK3Q4nXKEWdXzpFp0uVKN7xjYSbQB6+pOvHF2qHIKwlmvkdf9whdD9r+GGq9jq) --jasypt.encryptor.password=aWRhc2hib2FyZFNIY3VyaXR5S2V5 --`

```
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.schedul
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.schedul
r.JobSchedulerImpl.beforeJob - Before Job process.....!
```

Figure 18: Collectors

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

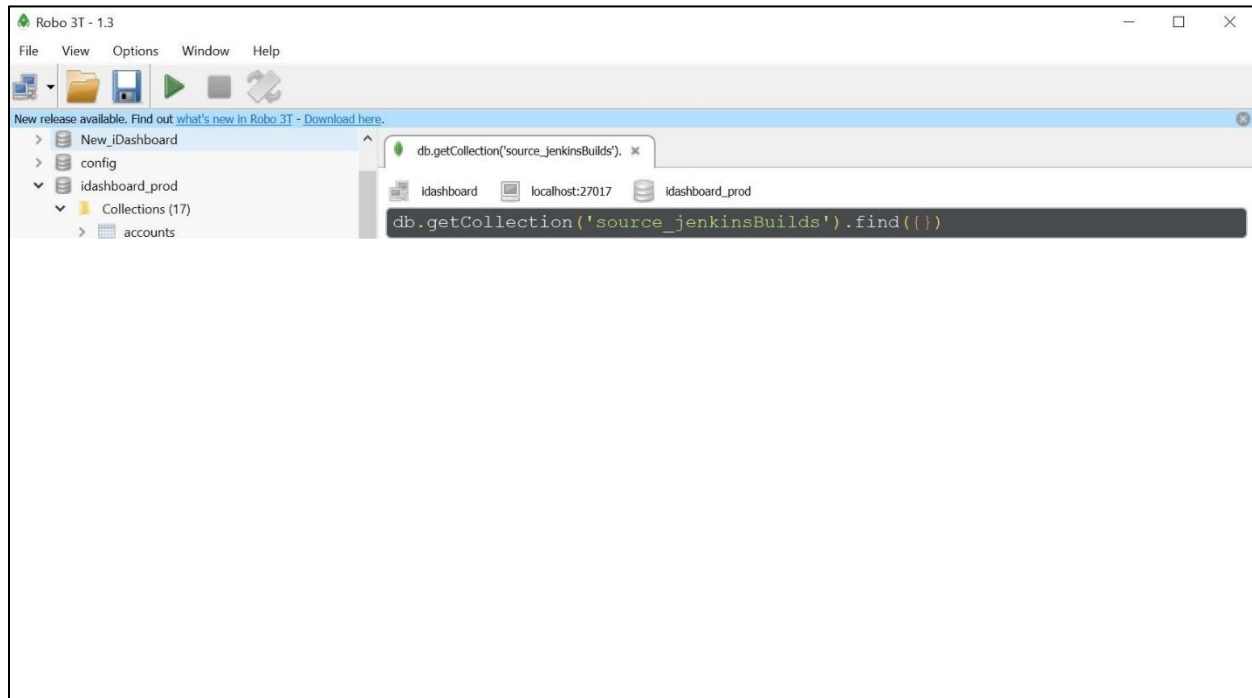



Figure 19: Refresh and verify

Note

Jasypt.encryptor.password as in **step no: 6** of the section [Encrypt Password](#)

6.2.5. NGINX Configuration

Follow the steps below after downloading the software. Refer the section [Software Requirements](#) for downloading NGINX.

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

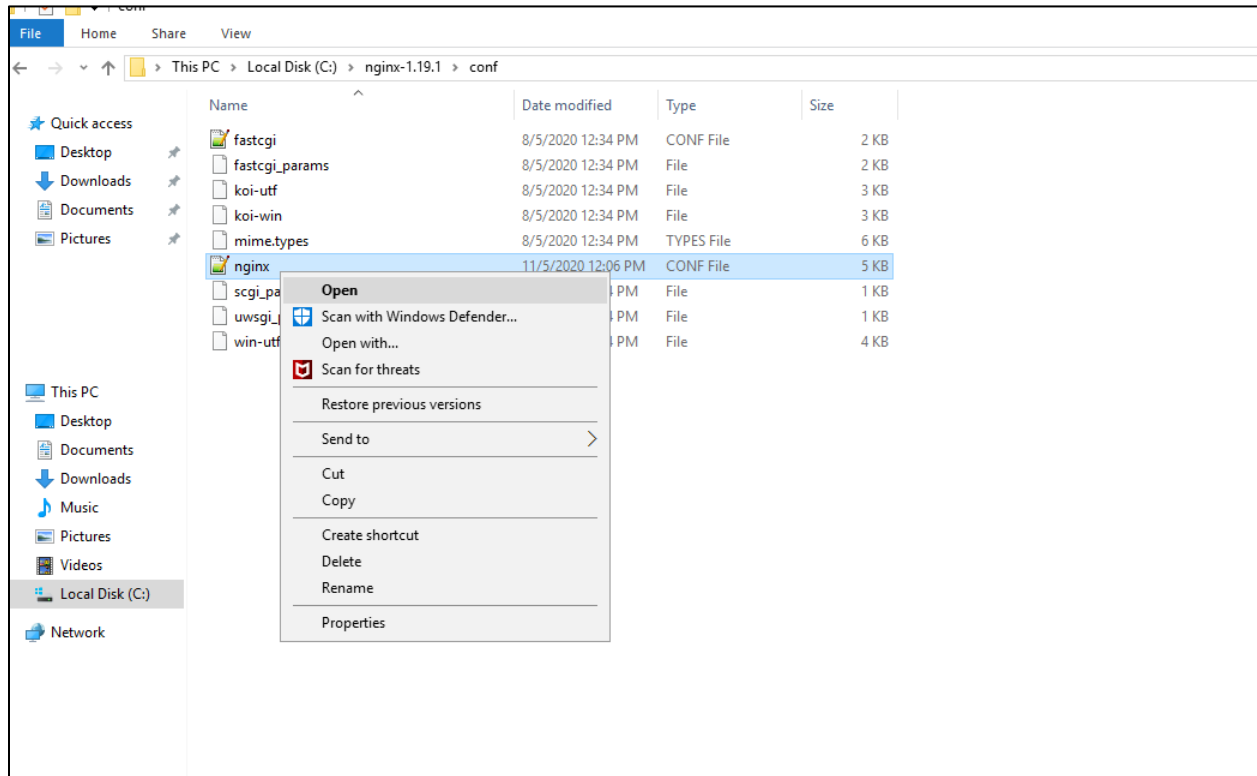


Figure 20: nginx.conf

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

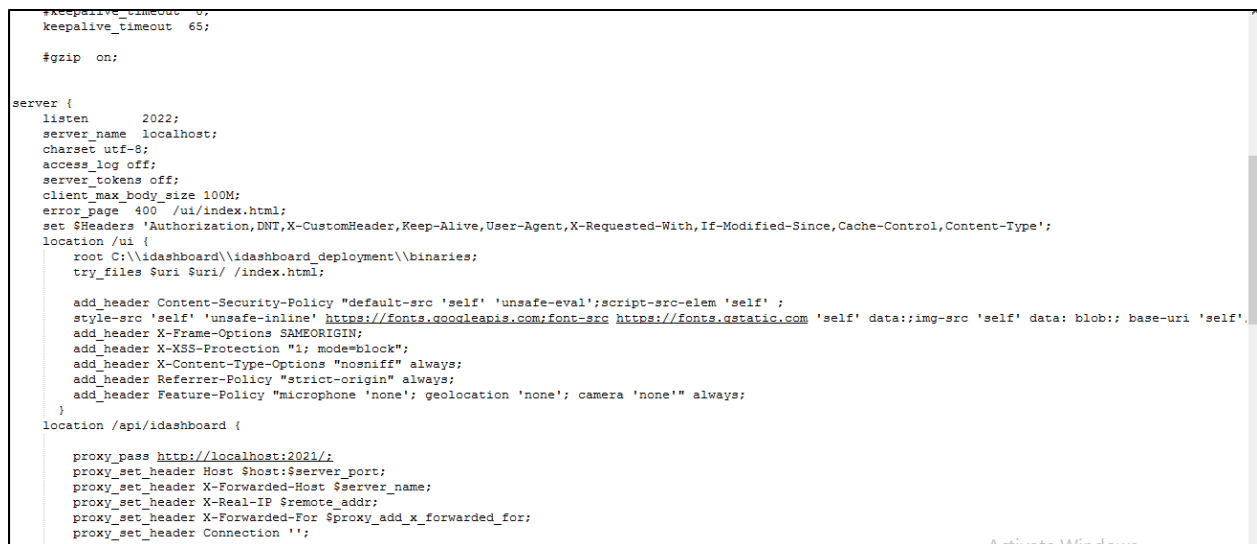


Figure 21: listen port

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

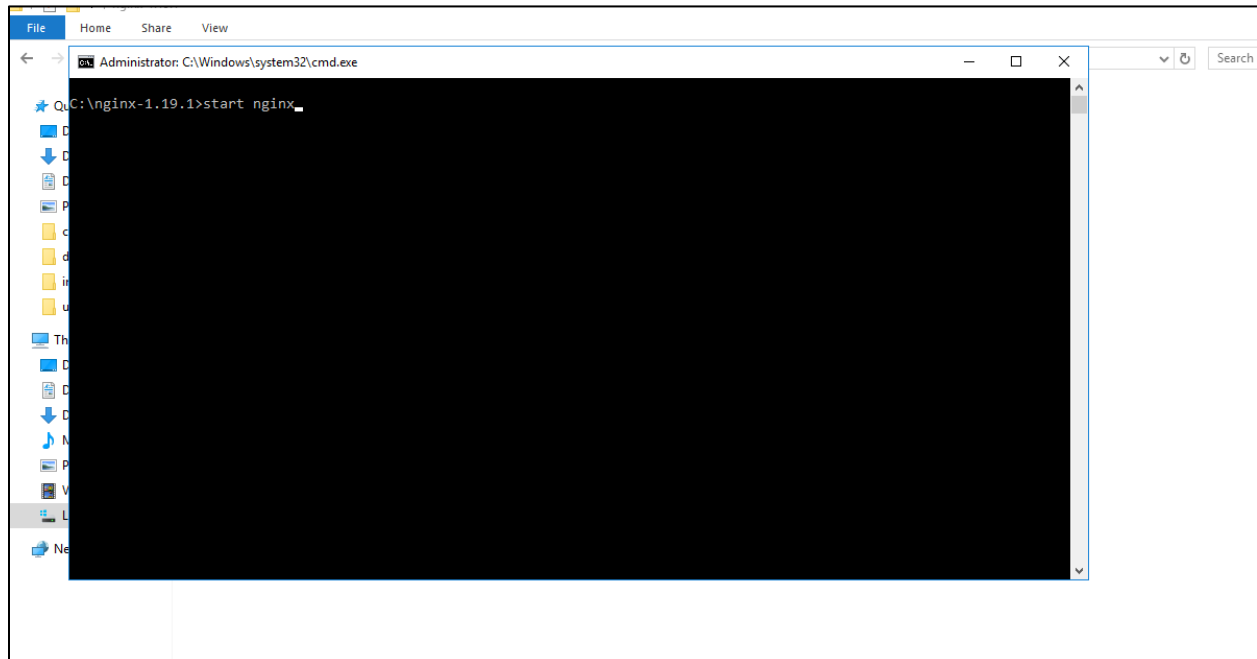


Figure 22: start nginx

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

6.2.6. Apache server Configuration

Follow the steps below after downloading the software. Refer the section [Software Requirements](#) for downloading Apache Server

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

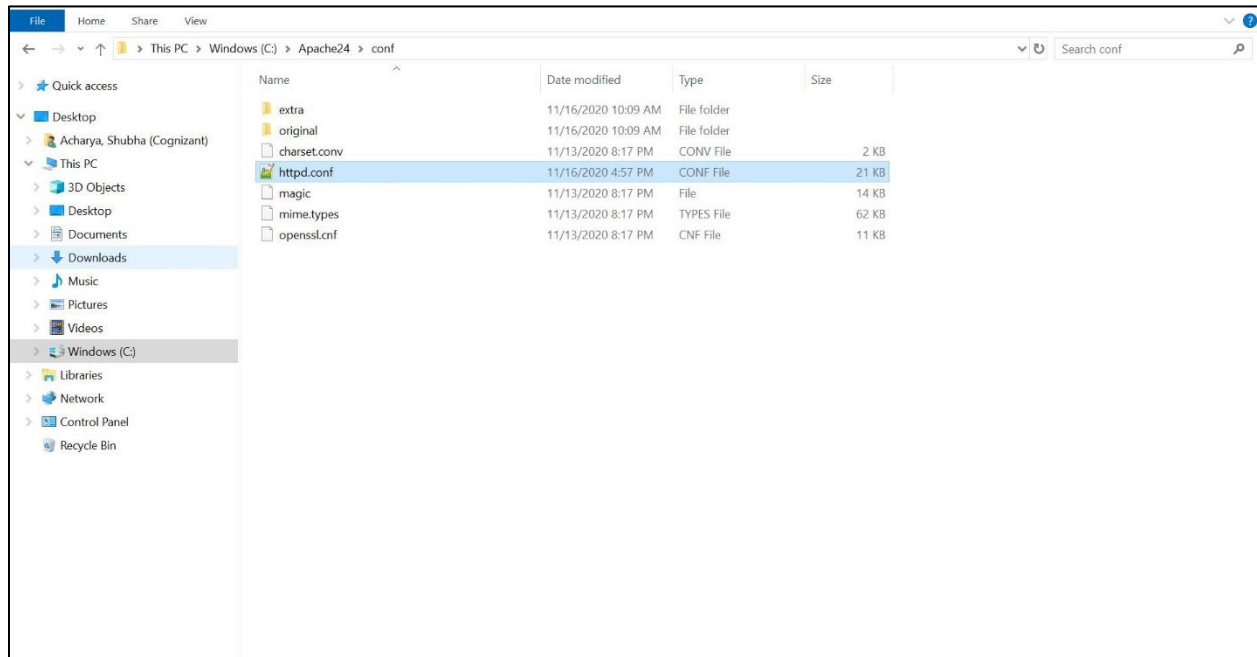


Figure 23: httpd.conf

- Open a command prompt window and cd to the c:\Apache24\bin folder.
- To Start Apache in the command prompt type:
`>httpd.exe`
- You can test your installation by opening up your Browser and typing in the address:
`http://localhost`
- You can shut down Apache by pressing Ctrl+C (It may take a few seconds)
- To install as a service. Open command prompt as Administrator and type:
`>httpd.exe -k install`
- You can start/stop the service with the command:
`>services.msc`

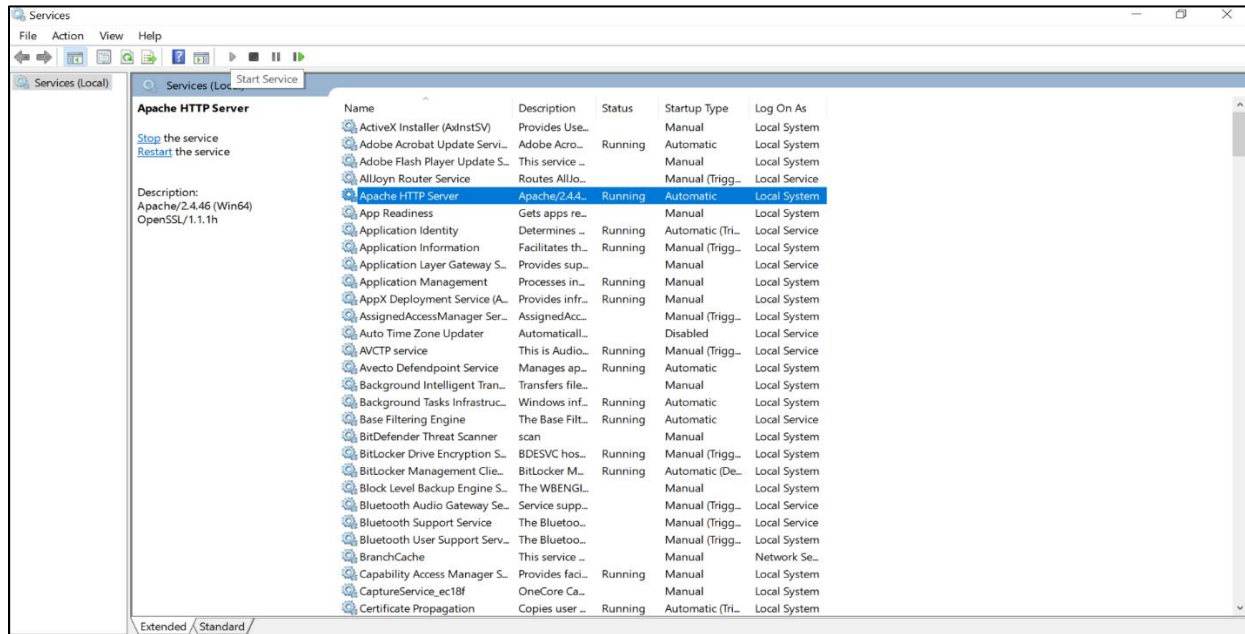


Figure 24: services.msc

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

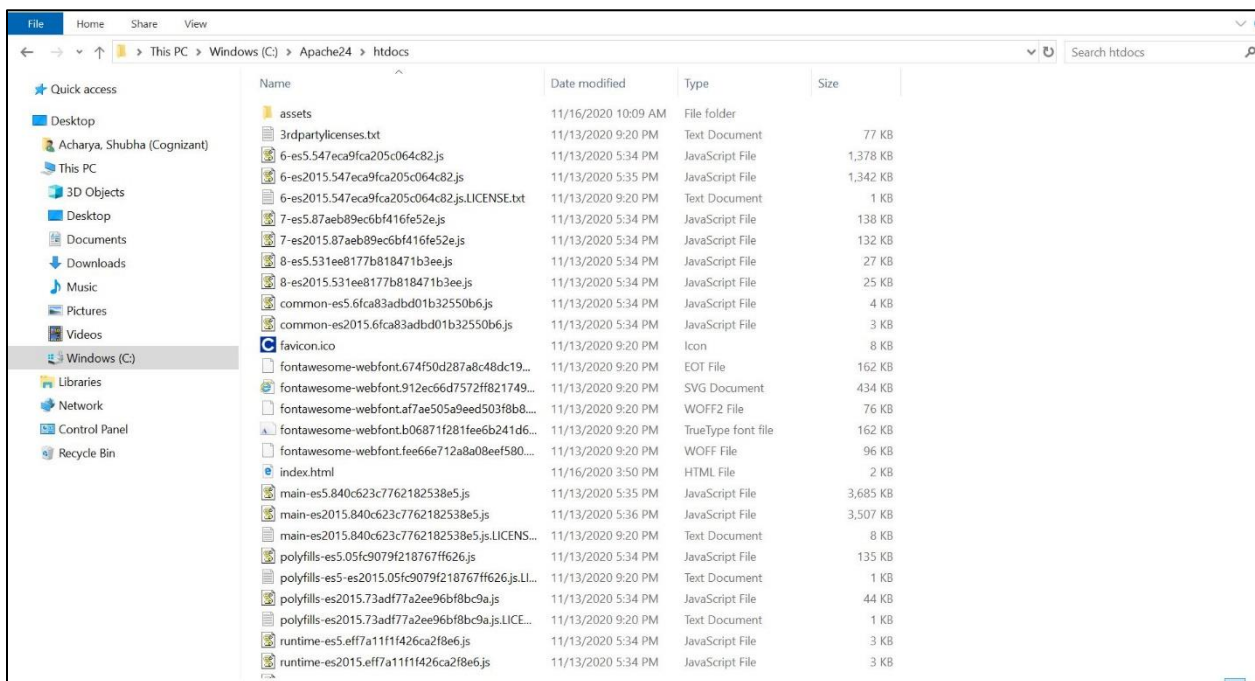


Figure 25: htdocs

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

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

Figure 26: httpd.config

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

7. FAQs

- 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](#))
- How can I change the database name?
To change the db name, use **--spring.data.mongodb.uri**
mongodb://<username>:\${spring.data.mongodb.creds}@localhost/<database-name> =
(Refer section [6.2.2](#) ,[6.2.3](#) and [6.2.4](#))
- 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