cQube – Operational Details Document

Oct 2020

Current Version 1.5

Document released by:
Sreenivas Nimmagadda

Document reviewed by:
Radhika Prabhu

Document acceptance by:
Arvind Gopalakrishnan

Version History

Release Version	Release Date	Covered Topics	Affected Sections
V 1.0	04.06.2020	• One step Installation • Data emission API • Role based login for Admin, Report viewer and Emission users • Change Password • Create New User • Student Attendance Report • CRC App data Report • Semester Assessment Report	Document was created with the below sections. 1. Purpose of this document 2. Background 4. cQube Requirements 6. cQube - Product Installation 7. Data emission process 8. Database Structure 9. cQube NIFI data file processing and data validations 12. Reports
V 1.1	16.07.2020	• Infrastructure reports with configuration stage implementation • API development for downloading log files and files, metrics from S3 buckets • Data validations • Admin screens • Access management using Keycloak	5. cQube network setup 9.1.4 Data Validation after Ingestion 10. Admin login process & Features 11. cQube release upgradation process 12.2 Scatter plot visualization
V1.2	20.08.2020	• Diksha data content play visualization • Telemetry implementation and telemetry visualization • Enhancement of the process scheduler • cQube Configurations • Keycloak integration automation • User management through admin screens	3. Prerequisites for cQube installation 6.2 cQube - Configurations 10.1.1 Admin - Features - Point 3 10.1.1 Admin - Features - Point 1 12.3 Stacked bar report
V1.2.1	26.08.2020	• Overall usage count for textbooks, courses in Diksha charts, Metrics for	12.4 Table format visualization 12.5 Bar Chart visualization

		diksha usage count, Diksha report in summary statistics ● crc report metric issue	
V1.3	11.09.2020	● UDISE data map based visualization	4.1 cQube product prerequisites Updated 6.2 cQube - Configurations Updated 6.3 Workflow Updated 11.3 Configurations at cQube upgradation process 12. Reports Updated
V1.4	30.09.2020	● Composite report configuration ● Scatter plot visualization using the composite data	6.2.1 Enable / Disable the process groups 6.2.5 U-DISE configuration 6.2.6 Composite report configuration 7.1.2 Emission order 7.2 Emission file naming conventions & structure 7.2.1 static File Name Structure 12. Reports 12.5 Bar Chart visualization
V1.4.1	09.10.2020	● Login page with new design ● Landing page with new design ● Percentage representation at Diksha column chart	
V1.5	16.10.2020	● UDISE Report - Configuration stage implementation ● DIKSHA api call ● Telemetry dashboard implementation ● PAT report enhancement - addition of time range	6.2.5 U-DISE configuration

Table of Contents

1. Purpose of this document

The purpose of this document is to describe the operational details of the cQube product. The operational details include description of functionalities like the cQube product installation, admin functionalities and upgradation of the cQube product. Some of the technical details like the cQube network setup and the data emission process have been added to this document to understand the operations clearly.

2. Background

cQube is an analytics product for the education system, this product can be used for monitoring the education system in a state and on a broader scale for monitoring the schools across the state/ districts/ clusters/ blocks/ villages and schools.

2.1 Use of this document

This document can be used by anyone who works with the cQube product and whoever has the permission to view the documents of the completed project. This document helps the user understand the design and architecture, as well as the operational and technical details of the cQube.

2.2 State of this document

This document is in a final draft version and is under change control. To request a change to the technical document please contact the system architect or the project manager.

2.3 Acronyms

The following is a list of acronyms which will be used throughout this document:

Table – 1: Acronyms

Acronym	Description
S3	Simple Storage Service
NIFI	Apache NIFI
PK	Primary Key
FK	Foreign key
RDAC	Restricted Database Access Control

NSQF	National Skills Qualification Framework
PGI	Postgraduate Institute
CWSN	Children with special needs

3. Prerequisites for cQube installation

Mentioned below are the prerequisites for the installation of the cQube product:

- The cQube product has to be installed in the AWS environment
- Firstly an instance with Ubuntu 18.04 OS has to be created.
- All the hardware requirements mentioned in section 4.1.4 have to be adhered to
- Before starting installation, the network set-up has to be completed as mentioned in the section 5 of this document.

The below table describes the Infrastructure details to install the cQube

Activity	**Infrastructure used for cQube installation**	**Required Skills**
1-2 Weeks	2 Days	
Filing of the requirements in the required format and obtaining relevant approvals for procurement of the Infrastructure and securing funding for monthly/yearly spend.	AWS account 1. cQube Server (CPU - 8 Core RAM - 32 GB Storage - 500 GB) 2. S3 Buckets (S3 emission 100GB S3 input - 750 GB S3 Output - 250 GB)	AWS Basic Operation Skills - EC2 - S3 - Load Balancer - IAM - VPC - Route53 - Certificate Manager (SSL)
	OpenVPN Access Server (EC2 instance CPU - 1 Core RAM - 2 GB Storage - 10 GB)	VPN admin Operations
	NGINX Reverse proxy(EC2 instance CPU - 2 Core RAM - 4 GB Storage - 30 GB)	NGINX configuration skills
	NAT gateway	AWS NAT gateway
	Ubuntu 18.04	Will be taken care while

		creating EC2 instance
	Java jdk 1.8	Will be installed through One-Step cQube Installation
	Python 3	Will be installed through One-Step cQube Installation
	NIFI 1.11.4	Will be installed through One-Step cQube Installation
	Angular 9.1.6, Chart.js 2.9.3, Leaflet 1.6.0	Will be installed through One-Step cQube Installation
	PostgreSQL 10.12	Will be installed through One-Step cQube Installation

The table below gives the estimated time for the cQube network setup and product installation

S No	**Task Details**	**Estimated Time(Approx.)**
1	To create the EC2 Instance with Ubuntu 18.04	5 mins
2	For overall network setup	30 mins - 40 mins
3	cQube Installation	30 mins
4	Data emission for Student attendance, CRC and Semester assessment data	1 min with AWS otherwise it depends on the internet speed
5	Data processing time for Student attendance, CRC and Semester assessment data, Infrastructure, Diksha, UDISE	150 -180 mins

For more details of the data processing time please refer section 6

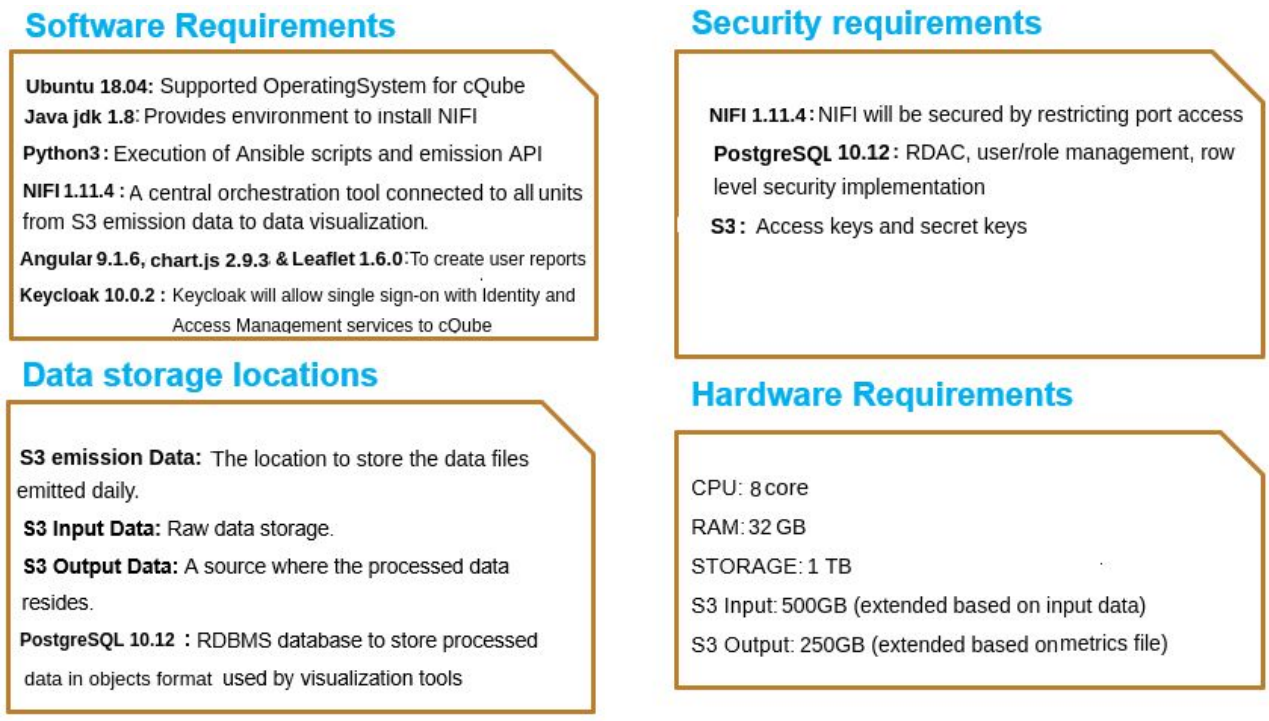
4. cQube Requirements

This section describes the prerequisites to install and configure the cQube product setup and the cQube product setup process. This section also describes the cQube network setup and the setup process

4.1 cQube product prerequisites

The cQube product installation process has a few system prerequisites that have to be followed. The system software, hardware, and security requirements have been derived and have to be adhered to before installation. The figure below gives an overview of the software requirements, security requirements, hardware requirements and the data storage locations:

Figure – 1: System hardware, software requirements, security requirements and data storage



4.1.1 Software Requirements

- Ubuntu 18.04: This is the operating system that supports the cQube product
- Java JDK1.8: This provides an environment for NIFI installation

- Python3: Python plays a role in the execution of Ansible scripts and data emission API using a virtual environment.
- NIFI 1.11.4: A central orchestration tool which is connected to all the units and all the different sections where the data flows, starting from the S3 emission data location to the data visualization stage
- PostgreSQL 10.12: An RDBMS database to keep all the processed data in relational data format. The data stored here is used by NIFI to prepare the JSON format files which can be used in the visualization charts.
- Angular 9.1.6 + ChartJS 2.9.3 + Leaflet 1.6.0: Angular, ChartJS and Leaflet are used to create dashboards/ user reports. The data stored in the S3 output bucket can directly be used to create the reports.

4.1.2 Data Storage Locations

- **S3 emission data bucket:** The data emitted from the state education system has to be stored until the NIFI reads the data. S3 emission input storage buckets are the storage locations where the emitted data files are stored.
- **S3 Input data bucket:** This is a location where the raw data resides for all future references.
- PostgreSQL: All the transformed and aggregated data will be stored in PostgreSQL tables.
- **S3 Output data bucket:** A location where the processed data resides in JSON format.

4.1.3 Software security requirements

- cQube product security will be provided by implementing the private subnet with AWS load balancer as described in the section 2 of this document.
- All the ports will be accessed by Nginx server only, So those ports will not be accessed directly from the internet.
- S3 buckets will be secured by the AWS default security.
- PostgreSQL will be secured by the following ways
  (The database security will be implemented in the future versions of cQube)
    - **RDAC:** Postgres provides mechanisms to allow users to limit the access to their data that is provided to other users. Database super-users (i.e., users who have `pg_user.usesuper` set) silently bypass all of the access controls described below with two exceptions: manual system catalog updates are not permitted if the user

does not have pg_user.usecatupd set, and destruction of system catalogs (or modification of their schemas) is never allowed.

- **User and Role Management:** the user roles will be granted with one or more cQube database will have three different types or roles:
    1. role role (identified by prefix r_)
    2. group role (identified by prefix g_)
    3. user role (generally personal or application names)
- **Row Level Security:** In addition to the SQL-standard privilege system available through GRANT, tables can have row security policies that restrict, on a per-user basis, which rows can be returned by normal queries or inserted, updated, or deleted by data modification commands. This feature is also known as Row-Level Security. When row security is enabled on a table all normal access to the table for selecting rows or modifying rows must be allowed by a row security policy.

4.1.4 Hardware requirements

Listed below are the minimum hardware specifications/ requirements in order to install the cQube product

- 8 core CPU
- 32 GB RAM
- 500 GB - 1 TB harddisk
- Three S3 Buckets will be used - for cQube data emission, cQube data input and cQube data output

  (Currently, the S3 buckets will be created automatically during installation. This will be revised in future releases and the AWS Admin will have to create the S3 buckets before starting the installation)

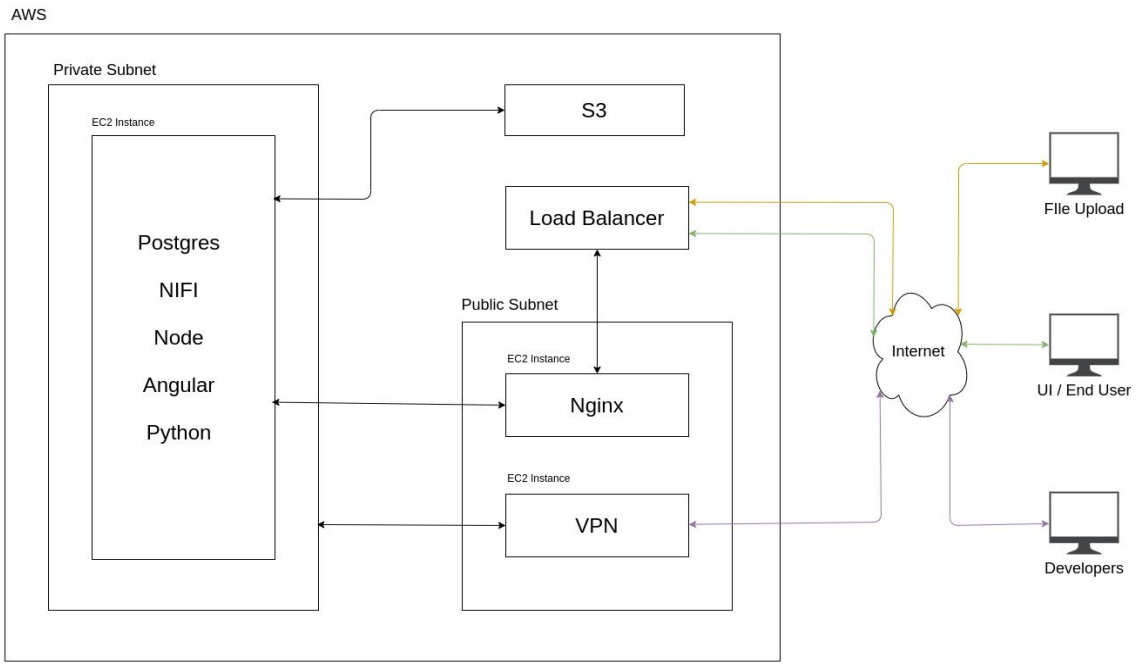
5. cQube network setup

The following steps define how the cQube setup and workflow is completed in AWS. cQube mainly consists of the areas mentioned below:

1. Private Subnet
2. Public Subnet
3. AWS Load Balancer
4. IAM user and Role creation for S3 connectivity.

The cQube network setup process is described in the block diagram below:

Figure – 3: cQube network setup diagram



- The Yellow arrows in the network diagram indicates the file upload users' connectivity through the load balancer.
- The Green arrow in the network diagram indicates the end users connectivity through the load balancer.
- The purple colour arrow in the network diagram indicates the developers connectivity through the VPN.

5.1 Private Subnet

Private subnet is used to secure the cQube server from public access. The instance will not be having the public ip. An EC2 instance will be created in a private subnet and all cQube components will be installed in this.

The steps involved to create EC2 instance in private subnet

- Create a virtual private cloud(VPC) in AWS
- Create a subnet in the created VPC with no Routing Table attached to Internet gateway
- Create an EC2 instance to install all the cQube software components.

EC2: cQube server

- create an ec2 instance with below mentioned configuration
- 8 core cpu, 32 gb ram and 1 tb storage

security_group:

- port 4200, 3000, 5000 inbound from nginx
- port 4201, 3001, 9000, 8080, 8096, 22 inbound from openvpn server

Load Balancer:

- create a load balancer
- configure it to connect to nginx on port 80, 443

Domain Name:

- create a domain name
- configure CNAME of load balancer to domain name

SSL:

- create a certificate from AWS Certificate Manager
- attach it to load balancer

Security Group:

- port 80, 443 inbound from 0.0.0.0/0

5.2 Public Subnet

Public subnet will contain two EC2 instances, one is for OpenVPN and another is for Nginx, which will act as a reverse proxy. It is used to provide connectivity with the private subnet.

The steps involved to create the public subnet

- Create a subnet in that same VPC where the private subnet was created with a Routing Table attached to the Internet Gateway.

- Create the first EC2 instance with OpenVPN AWS AMI and configure it to connect with the private subnet.
- Create the second EC2 instance with Ubuntu 18.04 AWS AMI and install Nginx to connect with the cQube server which is in the private subnet.

EC2: Openvpn server

- create an ec2 instance with openvpn ami
- create users to access the instances and for cQube admin pages

security_group:

- port 22 inbound to 1 public ip (admin's)

EC2: Nginx server

- create an ec2 instance with ubuntu 18.04 ami
- configure the nginx to connect Load balancer and cQube server
- configuration file will be provided

security_group:

- port 80, 443 inbound to Load balancer
- port 22 inbound to openvpn server

NAT Gateway:

- create a nat gateway / nat instance
- configure the connection to cQube server

security_group:

- port 80, 443 inbound from private_subnet

5.3 AWS Load Balancer

AWS load balancer is used in cQube to avoid the security risks and also to control the traffic among the servers to improve uptime and easily scale the cQube by adding or removing servers, with minimal disruption to cQube traffic flows. cQube is using the Classic Load Balancer

The steps involved to create the load balancer

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home
2. On the navigation bar, choose a Region for your load balancer. Be sure to select the same Region that you selected for your EC2 instances.
3. On the navigation pane, under LOAD BALANCING, choose Load Balancers.

4. Choose Create Load Balancer.
5. For Classic Load Balancer, choose Create.

The steps involved to configure the load balancer with EC2 instance

- On the Configure Health Check page, leave Ping Protocol set to HTTP and Ping Port set to 80.
- For Ping Path, replace the default value with a single forward slash ("/"). This tells Elastic Load Balancing to send health check queries to the default home page for your web server, such as index.html.
- On the Add EC2 Instances page, select the Nginx instance to register with the load balancer.

5.4 IAM user and Role creation for S3 connectivity

An AWS Identity and Access Management (IAM) user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS. A user in AWS consists of a name and credentials. An IAM user with **administrator permissions** is not the same thing as the AWS account root user.

To provide the connectivity between EC2 and S3 we need to create an IAM user with a supported role. The role should have the List, Read and Write permissions

We have multiple ways to create the IAM user account, But in cQube we created the IAM user from the AWS GUI by following the below points.

AWS S3:

- create 3 buckets for input, output and emission

IAM User:

- create an IAM user
- assign iam_policy to user
- download the aws access_key and secret_key

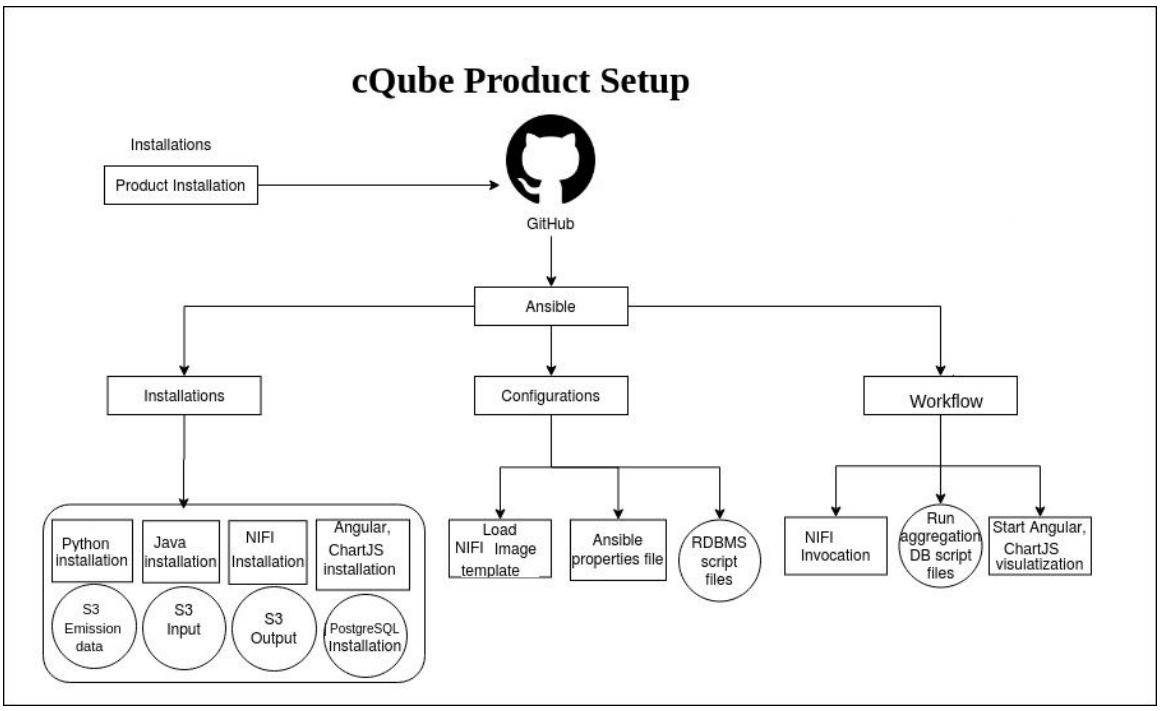
IAM Policy:

- create a policy from AWS IAM
- provide access to list, read and write the object to s3 buckets

6. cQube - Product Installation

The diagram below describes the cQube product installation process, the product configuration process and an overview of all the different software that have been used in the cQube tech stack.

Figure – 2: Flow diagram for the cQube product setup



6.1 cQube Installation

The cQube product can be installed with a one-step installation process. The one-step process installs the complete cQube stack, which includes Ansible automation scripts to install Java, Python, NIFI, Angular, ChartJS, Leaflet, S3 emission data, S3 input bucket, S3 output bucket and PostgreSQL installations.

Steps for Installation:

- As a first step, clone the files from GitHub using the following command:
  $ git clone https://github.com/project-sunbird/cQube.git.
- This downloads the cQube installation files. The cQube GitHub has all the installation files required for installing cQube.
- The README.md file has all the instructions that have to be followed for the cQube product installation.

- The install.sh script installs the complete cQube stack.
- The Install.sh file calls the Ansible playbooks in the background which will complete the cQube installation setup.
- The complete installation process takes approx 30 minutes.

6.2 cQube - Configurations

Configurations are done as a part of the installation process by the ansible scripts. All the properties of the sensible information like URLs and passwords are saved in the ansible properties file which is encrypted by default. Configuration is an automated process, no manual interactions are required in this stage.

cQube configurations through Ansible code:

- Nifi template would be uploaded & instantiated with the variables & parameters.
- All the Nifi controllers will be enabled and all the Nifi processors will be started.
- Postgres static, transaction, aggregation tables will be created in the newly created database.

The below configurations should be done in cQube while the installation and the data process.

- Enable / Disable the process groups - At installation time
- Keycloak two factor authentication - Need to verify for the roles Admin and Report viewer togo with 2 factor authentications at the time of login.
- Nifi process configurations - Change the query parameters for Nifi process and verify whether those are affecting the process or not
- Infrastructure configuration - Need to check if the selected infra values are affecting or not.

The above four configurations in explanation,

6.2.1 Enable / Disable the process groups

This configuration should be performed at the installation / Up-gradation stage of cQube. The cQube user may select the process groups which they wanted to include into the cQube. While the installation / Up-gradation of the cQube user will be available to select the existing process group by giving the true / false in the datasource_config.yml document which is available at the below link.

https://github.com/project-sunbird/cQube/blob/release-1.2.1/ansible/installation_scripts/datasource_config.yml

All the configuration should be given like below,

nifi_telemetry: true

nifi_crc: true

nifi_attendance: true

nifi_semester: true

nifi_infra: true

nifi_diksha: true

nifi_udise: true

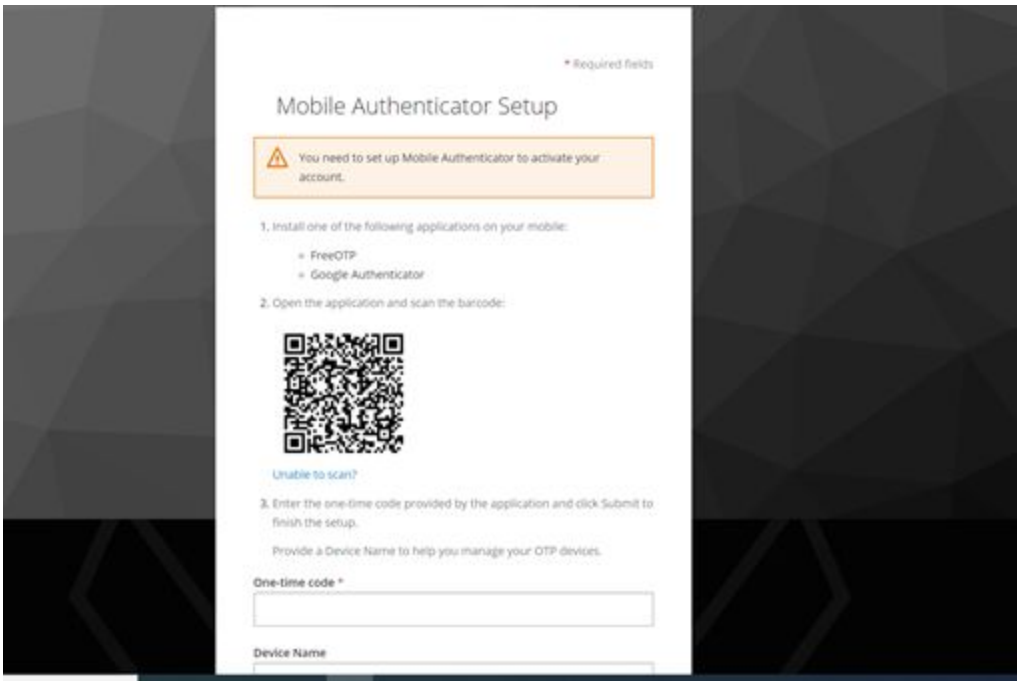
nifi_pat: true

nifi_comp: true (If composite report is required)

As of now there were a total of 7 processor groups available in cQube.

6.2.2 Keycloak Two-Factor Authentication

Keycloak will allow single sign-on with Identity and Access Management services to cQube. Keycloak will be configured to cQube at the time of installation in the execution process of install.sh command. If two factor authentication is set up the below screen would appear after first time login for all the users. Users need to set up two factor authentication for the first time from his/her mobile using the Google authenticator.



Below are steps:

- Download Google Authenticator app into your mobile.

- Scan the QR code to integrate cQube dashboard with Google authenticator app.
- Provide the OTP to login.

6.2.3 Nifi query parameter configurations

The Nifi configuration process allows us to change the query parameters before the installation process starts. To change the query of a nifi processor, we need to update the query parameter of that processor.

- Query should be re constructed outside the cQube.
- Query parameters should be defined in the configuration files of the nifi processor group.
- Additional filters can be added in the query parameter.
- Cast, round other functions can be used to update the query in the parameter configuration.
- The query in the NIFI process should be replaced with the new Query.
- The Query results should be affected on UI Reports.
- Below are a few examples to add or remove filters & apply functions to the query.

Example:

1. Adding a filter to the existing query for the static processor

By default the static_get_invalid_names parameter in static_data_parameters.txt would be

The Original Query:
"static_get_invalid_names":'"select school_id,school_name,block_id,district_id,cluster_id from school_hierarchy_details where cluster_name is null or block_name is null or district_name is null

The Updated Query:
"static_get_invalid_names":'"select school_id,school_name,block_id,district_id,cluster_id from school_hierarchy_details where cluster_name is null or block_name is null or district_name is null or school_name is null"',

Output: By adding the additional filter to not allow the school_name with null records into cQube reports.

2. Updating infra parameter based on active infrastructure attributes
By default the infra_normalize parameter in infra_parameters.txt would be

The Original Query parameter:

"infra_normalize":'"select school_id ,
case when HaveDrinkingWater <>1 then 0 else 1 end as drinking_water,
case when NoOfToilet=0 or NoOfToilet is null then 0 else 1 end as toilet,
case when HaveCWSNToilet <>1 then 0 else 1 end as cwsn_toilet,
case when HaveElectricity <>1 then 0 else 1 end as electricity,
case when HaveCCTV <>1 then 0 else 1 end as cctv,
case when HaveLibrary <>1 then 0 else 1 end as library from flowfile"'

The above query parameter can be changed according to the data fields activated by the state.

If only three of the fields choose to be activated by the state the query needs to be updated as below.

The Updated Query parameter:

```
"infra_normalize":'"select school_id ,
case when NoOfToilet =0 then 0 else 1 end as toilet,
case when HaveElectricity<>1 then 0 else 1 end as electricity,
case when solarpanel_yn=TRUE then 1 else 0 end as solar_panel from flowfile"'
```

We need to update the query in the infrastructure configuration file i.e., infra_parameters.txt file after that the installation process can be started. This will map with the infrastructure input data with the cQube database tables.

6.2.4 Infrastructure configuration

cQube is having the flexibility to support multiple state data with minimal changes in UI code for the fluctuated data sources like Infrastructure, Infrastructure score weights, Semester assessment subjects.

Below Steps will be performed to implement the database configuration stage:

- Before cQube is installed, the infrastructure data source needs to be configured depending on the emission data fields required by the state in the infrastructure configuration file.
- During configuration users can activate/deactivate the data fields, but not the key columns (ex : student_id,school_id cannot be deactivated). Fields like infrastructure attributes for particular states can be activated/deactivated.
- In configuration stage fields can be activated/deactivated based on requirement of state.
- Depending on the infrastructure attributes and its datatype the infrastructure query parameter, a case statement is written to map the input data field and the cQube infrastructure table.
- The case statement needs to be updated in the infrastructure configuration file infra_parameters.txt file by updating the parameter: infra_normalize.
- From the below table the infra_normalize parameter will look like :

  ```
  "infra_normalize":'"select school_id ,
  case when NoOfToilet =0 then 0 else 1 end as toilet,
  case when HaveElectricity<>1 then 0 else 1 end as electricity,
  case when solarpanel_yn=TRUE then 1 else 0 end as solar_panel from flowfile"',
  ```

- Example table :

datatype	input data	infrastructure_master.csv	case statement
integer	NoOfToilet	Toilet	case when NoOfToilet =0 then 0 else 1 end as toilet,
bit	HaveElectricity	Electricity	case when HaveElectricity<>1 then 0 else 1 end as electricity,
boolean	Solar Panel	Solar Panel	case when solarpanel_yn=TRUE then 1 else 0 end as solar_panel,

Note :- For infrastructure_master column the values should be converted to lowercase and spaces are converted to underscore(_) example : Solar Panel converted to solar_panel

- Based on the configuration, the changes are made to handle state specific data fields
- After successful validations, data tables are created with only active and required data fields for the active data fields are processed and metrics are generated
- The metrics generated will be stored in JSON files, which is used for visualization
- Change only the status of the attributes in infrastructure_master.csv, Don't update any other column values.

To select the infrastructure fields, please fill the details in infrastructure_master.csv file which is available at the link below.

https://github.com/project-sunbird/cQube/blob/release-1.3/ansible/installation_scripts/infrastructure_master.csv

The Infrastructure calculation will happen as in the below example.

If a school has drinking water, handwash, electricity, toilet, playground, hand pumps, library then the score would be calculated as below

20*1 + 10*1 + 10*1 + 20*1 + 20*1 + 10*1 + 10*1 = 100

The total infrastructure score would be 100, if the school does not have any infrastructure available, then for that infrastructure it would be awarded with 0.

If the school does not have playground, handpump, library, the score would be calculated as

20*1 + 10*1 + 10*1 + 20*0 + 20*1 +10*0 + 10*0 = 60

The total infrastructure score would be calculated to 60.

If we need to calculate the score at different levels such as district, block, cluster, the school infrastructure count would be added for all the blocks, clusters, districts.

To get the infrastructure score at district, block, cluster add all the infrastructure of all the schools available.

Weight * (infrastructure of all the school/ Total number of schools)

For example, if there are two schools in cluster the infrastructure score would be calculated as

20*(1+1)/2 + 10*(1+0)/2 + 10*(1+1)/2 + 20*(0+0)/2 + 20*(1+1)/2 +10*(0+0)/2 + 10*(0+0)/2 = 55

Infrastructure score for the cluster would be 55.

The Metrics will be stored in the JSON files for infrastructure visualization. An example JSON document for the district infrastructure is given below:

{"district": {"id": 2401,"value": "Kachchh"},"block": {"value": "Abdasa","id": 240107},"infra_score": {"value": "79.00"},"average": {"value": "170","percent": "79.81"}, "total_schools": {"value": 213},"total_schools_data_received": {"value": "213"},"handwash": {"value": "213","percent": "100.00"},"solar_panel": {"value": "203","percent": "95.31"},"library": {"value": "210","percent": "98.59"},"drinking_water": {"value": "13","percent": "6.10"},"tap_water": {"value": "13","percent": "6.10"},"hand_pumps": {"value": "158","percent": "74.18"},"playground": {"value": "159","percent": "74.65"},"news_paper": {"value": "213","percent": "100.00"},"digital_board": {"value": "3","percent": "1.41"},"electricity": {"value": "207","percent": "97.18"},"toilet": {"value": "213","percent": "100.00"},"boys_toilet": {"value": "213","percent": "100.00"},"girls_toilet": {"value": "213","percent": "100.00"}}

- Once the installation is completed, to configure the infrastructure weights, the infrastructure_score.csv file needs to be emitted. This activity should be done before the infrastructure data source emission.
- Code snippet to download the active infrastructure data fields file. After executing the code snippet infrastructure_score.csv file would be downloaded.
- Update the infrastructure scores in the downloaded infrastructure_score.csv file based on the state requirement.
- Once score is modified, execute the code to update the infrastructure weightage.

Note:-

1. While emitting infrastructure input data ,emit only the activated columns as in infrastructure_master.csv and only active infra's case statements should be updated in infra_parameters.txt and this has to be done before the emission of infrastructure data files.

6.2.5 U-DISE configuration

cQube is having the flexibility to support configuring multiple indices and its metrics with minimal changes in UI code for udise indices and metrics, udise score weights and there are 32 input data files updated in 7.2 section of this document that need to be emitted to visualize the UDISE report.

Below Steps needs to be performed to implement the UDISE configuration stage:

- Before cQube is installed, UDISE data source needs to be configured depending on the emission data fields required by the state in the udise configuration file.
- During configuration users can activate/deactivate the indices and metrics status and their corresponding weights, but not the key columns (ex : id,description,column,type,indice_id should not be modified/edited). Only fields like status and score are updated based on the active indices,metrics and their weights.
- In configuration stage fields will be activated/deactivated for that state, based on requirement
- We can also create our own indices and metrics during the udise configuration stage.
- There is an exhaustive list of calculated metrics available by default in cQube which is calculated from the UDISE raw input tables.
- The exhaustive calculated metrics can be used for creation of a normalised metric.
- We can create any number of normalised metrics from the exhaustive list by choosing to add any of the calculated metrics.
- The normalised metric can be defined under
    1. Newly created index with new normalized metrics
    2. Newly created index with few new normalized metrics and few existing normalized metrics with different name and metric_id.
    3. Existing index with new normalized metrics
- After successful validations the active metrics and indices are generated.
- The metrics/indices generated will be stored in JSON files, which is used for visualization
- There are 3 types of directions and they are No, Forward, Backward, based on the metric the direction can be configured. Based on this normalization takes place between 1 to 0 for backward metrics and between 0 to 1 for forward metrics.
- If the schools are not having any particular metrics, the total weights(denominator) are considered for the available schools. Here is an example in the Metric level configuration sheet.

Check udise_config_example sheet for examples of above three possible configurations.

To select the Udise indices,metrics please fill the details in the udise_config.csv file which is available at the link below.

https://github.com/project-sunbird/cQube/blob/release-1.5/development/postgres/udise_config.csv

Note :-

1. For providing weights the sum of score of all active indices should be 100
2. And also the score of all active metrics of each active indices should be 100
3. Change only **status** and **score** column from the udise_config.csv file, depending on the use case.
4. While opening udise_config.csv file in excel , please use '|' as a delimiter.

The indices,metrics normalization and calculations will happen as in the below example.

If a state selected community participation and medical index indices and their metrics as below

id	description	column	type	indice_id	status	score
3000	**community participation**	**Community_Participation**	**indice**		**1**	**60**
3001	% SMC members provided training	cp_smc_members_training_provided	metric	3000	1	40
3002	Total meetings held by SMC	cp_total_meetings_held_smc	metric	3000	1	40
3003	SMDC in school	cp_smdc_school	metric	3000	1	20
7000	**medical index**	**Medical_Index**	**indice**		**1**	**40**
7001	Medical Check-up Conducted	med_checkup_conducted	metric	7000	1	60
7002	De-worming Tablets	med_dewoming_tablets	metric	7000	1	20
7003	Iron Tablets	med_iron_tablets	metric	7000	1	20

The total infrastructure score will be calculated using below formulas :

1. Before start doing calculation, we will normalize the data by grouping them in quartiles as in the following document ,udise metric calculation logic

2. Community participation (cp)= (cpm1*40)/100+(cpm2*40)/100+(cpm3*20)/100
3. Medical index (med)= (medm1*60)/100+(medm2*20)/100+(medm3*20)/100
4. infra_score=(cp*60)/100+(med*40)/100

6.2.6 Composite report configuration

cQube is having the facility to combine the information of all reports and showing it in a single scatter plot and such report is called Composite report, Composite report is the combination of multiple reports which is used to co-relate the information between data sources in school, cluster, block, district level. Example : We can compare the attendance and semester performance of the school in a single scatter report.

1. This configuration should be performed at the installation / Up-gradation stage of cQube.
2. The user may select the process groups which they wanted to include into the cQube (As in 6.2.1) . The metrics available in the selected processor groups will be used in the composite report.
3. If composite report is required to be enabled, make sure the [nifi_comp: true] processor group is enabled in composite_enable similar to as in 6.2.1

6.3 Workflow

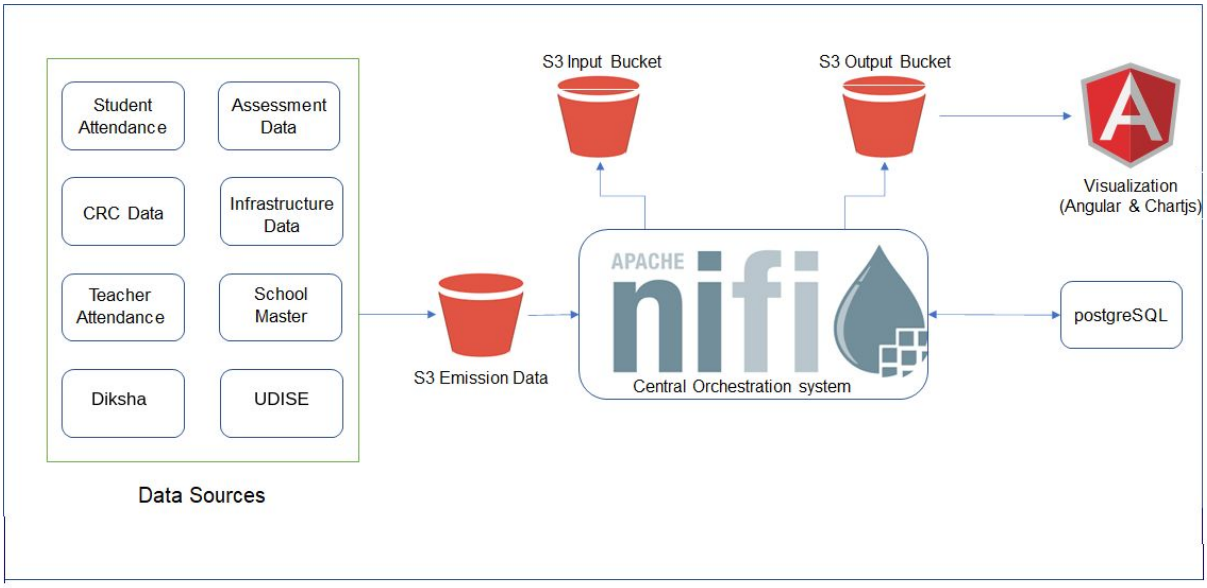
cQube workflow will be done in two parts

1. Data emission
2. Data processing

Data emission: Data from the state education system will be emitted into the S3 emission data folder using the cQube data emission API
emission API is described in the section 7

Data processing:

NIFI is the central orchestration system which handles the complete workflow process and generates JSON files. These JSON files are used for creation/ generation of the visualization reports. Data from the state education system will be emitted into the S3 emission data folder using the cQube data emission API. The below image will give more clarity on the data processing.



7. Data emission process

cQube provides a Rest API for the data ingestion. Authenticated API call provides onetime S3 URL and emits the data into the S3 emission bucket.

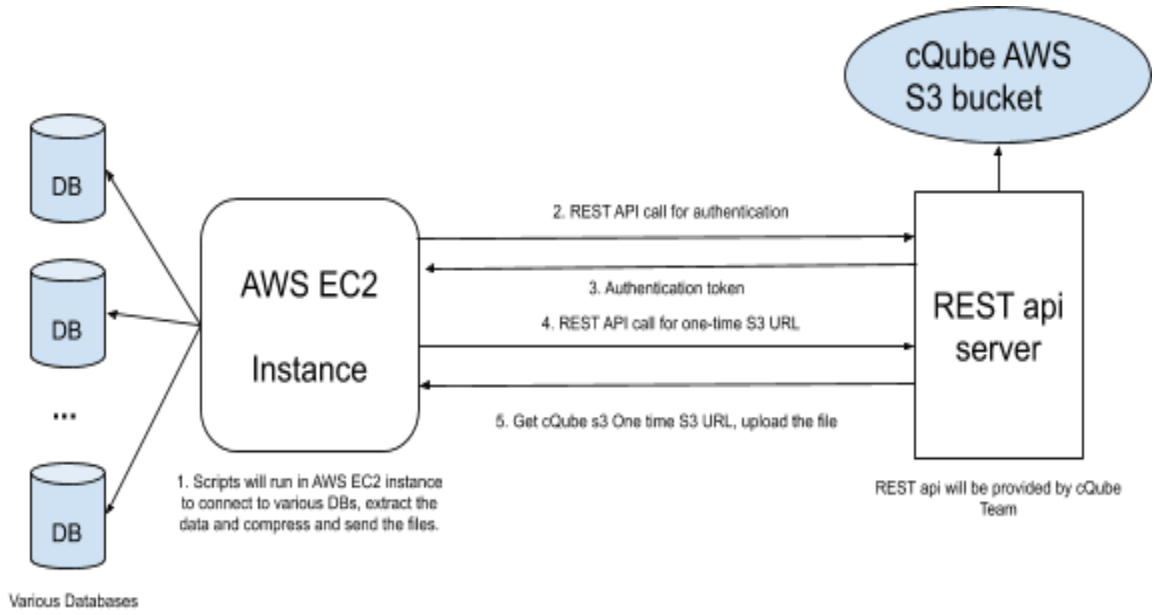


Figure – 4: Data emission process

7.1 The steps involved in the data Emission Process

- The data emission users will work from the data source location.

- Data emission users develop the automation extraction process which is outside cQube to extract the data from the sources.
- Data emission will be performed periodically as per the specified time interval from different data sources with the help of the above automated extraction process.
- Once the automation extraction process extracts the data fields which are required by cQube(Mentioned in section 4.2), it is converted into csv pipe delimited formatted files.
- The CSV data files will then be placed into the state data center by the automation extraction process.
- Emission process will be invoked by the automation extraction process once the extraction is completed.
- The emission process will be authenticated, An emission API will be invoked by authenticating the API token.
- The emission process makes an API call to generate AWS S3, one time presigned URL.
- API calls with the data file as a parameter to emit the data files using the https protocol into cQube.
- The emission details will be captured by the log files and will be available to the admin for review.

7.1.1 Emission APIs

- Data into cQube can be ingested through emission API.
- After extracting the data from the source, save the file with the data source name. Below are the naming conventions.
- For configuring the emission job, fill the details in config.py like emission URL, password, file path which is available in the configuration file at below location cQube/development/python/config.py
- After configuration to emit the files into the cQube system, invoke the cQube/development/python/**client**/client.py program.

7.1.2 Emission order

- Static files which are defined in section 7.2.1 in this document need to be emitted first.
- For the CRC report, the inspection_master needs to be emitted prior to user_location_master to visualize the CRC reports.
- Similarly for PAT (Periodic assessment report), periodic_exam_mst and periodic_exam_qst_mst needs to be emitted prior to periodic_exam_result_trans

7.2 Emission file naming conventions & structure:

- All the files should be in a CSV format with PIPE("|") separated.
- The data files should be emitted individually in zip format.
- The list of columns which have to be emitted for a data source are provided in the file.
- The header should be in the same order as described in the file, for that respective data set.
- Data emission overview of data flow can be viewed here
- Below is the list of data sources used in various reports.

Source Name	Used in report	cQube Data Source Name
District Master	All reports	district_master
Block Master	All reports	block_master
Cluster Master	All reports	cluster_master
School Master	All reports	school_master
CRC Inspection Master	CRC Report	inspection_master
CRC User Location Master	CRC Report	user_location_master
Student Attendance	SAR Report	student_attendance
Infrastructure Master	Infrastructure Report	infra_trans
Semester	SEMESTER Report	semester
Diksha	DIKSHA	diksha
School Facility	UDISE	sch_facility
NSQF Class 12 Placement	UDISE	nsqf_plcmnt_c12
NSQF Class 10 Placement	UDISE	nsqf_plcmnt_c10
NSQF Class Condition	UDISE	nsqf_class_cond
School Exam Result Class 12	UDISE	sch_exmres_c12
School Exam Result Class 10	UDISE	sch_exmres_c10
School Exam Result Class 5	UDISE	sch_exmres_c5

School Incentives CWSN	UDISE	sch_incen_cwsn
School Enrollment by stream	UDISE	sch_enr_by_stream
School Enrollment by CWSN	UDISE	sch_enr_cwsn
School Enrollment by Medium of Instruction	UDISE	sch_enr_medinstr
School Enrollment by Age	UDISE	sch_enr_age
School Enrollment by Report	UDISE	sch_enr_reptr
School Enrollment by Fresh	UDISE	sch_enr_fresh
School Enrollment by New Admission	UDISE	sch_enr_newadm
School staff position	UDISE	sch_staff_posn
School Exam Class 10 marks	UDISE	sch_exmmarks_c10
School Exam Class 12 marks	UDISE	sch_exmmarks_c12
NSQF Exam Class 10 Results	UDISE	nsqf_exmres_c10
NSQF Exam Class 10 Results	UDISE	nsqf_exmres_c12
NSQF Training Provision	UDISE	nsqf_trng_prov
NSQF Faculty	UDISE	nsqf_faculty
School Exam class 8 results	UDISE	sch_exmres_c8
School Profile	UDISE	sch_profile
Teacher Profile	UDISE	tch_profile
School Receipt Expenditure	UDISE	sch_recp_exp
NSQF Basic information	UDISE	nsqf_basic_info
NSQF Enrollment by Caste	UDISE	nsqf_enr_caste
NSQF Enrollment by Sub section	UDISE	nsqf_enr_sub_sec
School PGI Indicators	UDISE	sch_pgi_details
School Safety	UDISE	sch_safety

School Incentives	UDISE	sch_incentives
Periodic Exam Master	Periodic Exams	periodic_exam_mst
Periodic Exam Question Master	Periodic Exams	periodic_exam_qst_mst
Periodic Exam Results	Periodic Exams	periodic_exam_result_trans

Naming conventions of the files and the file structure should be followed as mentioned below:

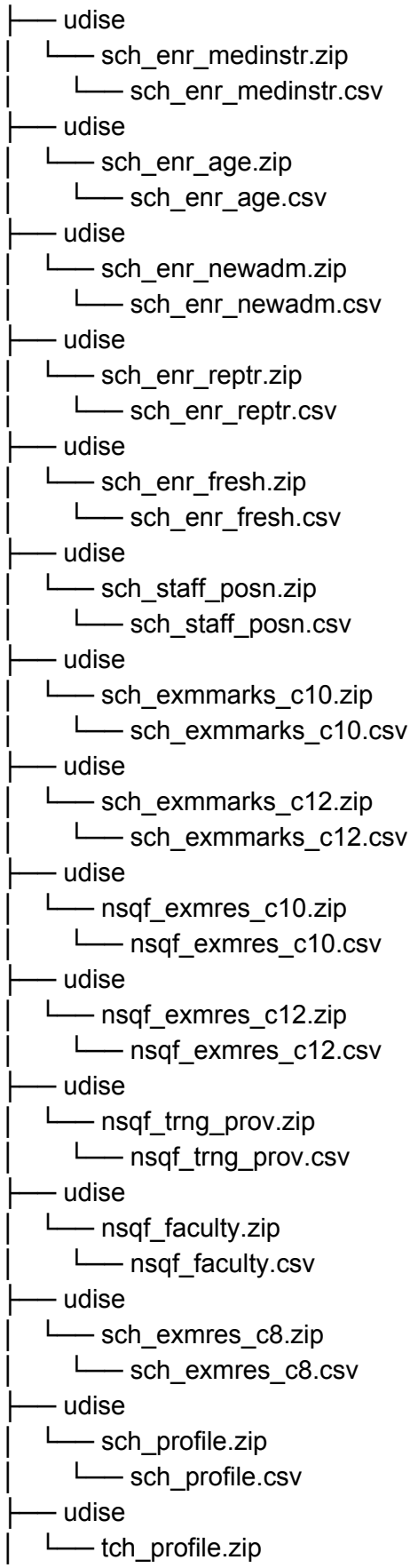
7.2.1 static File Name Structure:

```
cqube_emission
|
├── block_master
│   └── block_mst.zip
│       └── block_mst.csv
├── cluster_master
│   └── cluster_mst.zip
│       └── cluster_mst.csv
├── district_master
│   └── district_mst.zip
│       └── district_mst.csv
├── school_master
│   └── school_mst.zip
│       └── school_mst.csv
├── infrastructure_score
│       └── infrastructure_score.csv
├── pat
│   └── periodic_exam_mst.zip
│      └── periodic_exam_mst.csv
├── pat
│   └── periodic_exam_qst_mst.zip
│      └── periodic_exam_qst_mst.csv
|
```

7.2.2 Transactional File Name Structure:

```
cqube_emission
|
├── semester
│   └── semester.zip
│       └── semester.csv
├── student_attendance
```

```
│   └── student_attendance.zip
│       └── student_attendance.csv
├── user_location_master
│   └── user_location_master.zip
│       └── user_location_master.csv
├── inspection_master
│   └── inspection_master.zip
│       └── inspection_master.csv
├── infra_trans
│   └── infra_trans.zip
│       └── infra_trans.csv
├── diksha
│   └── diksha.zip
│       └── diksha.csv
├── udise
│   └── sch_facility.zip
│      └── sch_facility.csv
├── udise
│   └── nsqf_plcmnt_c12.zip
│      └── nsqf_plcmnt_c12.csv
├── udise
│   └── nsqf_plcmnt_c10.zip
│      └── nsqf_plcmnt_c10.csv
├── udise
│   └── nsqf_class_cond.zip
│      └── nsqf_class_cond.csv
├── udise
│   └── sch_exmres_c12.zip
│      └── sch_exmres_c12.csv
├── udise
│   └── sch_exmres_c10.zip
│      └── sch_exmres_c10.csv
├── udise
│   └── sch_exmres_c5.zip
│      └── sch_exmres_c5.csv
├── udise
│   └── sch_incen_cwsn.zip
│      └── sch_incen_cwsn.csv
├── udise
│   └── sch_enr_by_stream.zip
│      └── sch_enr_by_stream.csv
├── udise
│   └── sch_enr_cwsn.zip
│      └── sch_enr_cwsn.csv
```


udise
sch_enr_medinstr.zip
sch_enr_medinstr.csv
udise
sch_enr_age.zip
sch_enr_age.csv
udise
sch_enr_newadm.zip
sch_enr_newadm.csv
udise
sch_enr_reptr.zip
sch_enr_reptr.csv
udise
sch_enr_fresh.zip
sch_enr_fresh.csv
udise
sch_staff_posn.zip
sch_staff_posn.csv
udise
sch_exmmarks_c10.zip
sch_exmmarks_c10.csv
udise
sch_exmmarks_c12.zip
sch_exmmarks_c12.csv
udise
nsqf_exmres_c10.zip
nsqf_exmres_c10.csv
udise
nsqf_exmres_c12.zip
nsqf_exmres_c12.csv
udise
nsqf_trng_prov.zip
nsqf_trng_prov.csv
udise
nsqf_faculty.zip
nsqf_faculty.csv
udise
sch_exmres_c8.zip
sch_exmres_c8.csv
udise
sch_profile.zip
sch_profile.csv
udise
tch_profile.zip

```
│       └── tch_profile.csv
├── udise
│   └── sch_recp_exp.zip
│       └── sch_recp_exp.csv
├── udise
│   └── nsqf_basic_info.zip
│       └── nsqf_basic_info.csv
├── udise
│   └── nsqf_enr_caste.zip
│       └── nsqf_enr_caste.csv
├── udise
│   └── nsqf_enr_sub_sec.zip
│       └── nsqf_enr_sub_sec.csv
├── udise
│   └── sch_pgi_details.zip
│       └── sch_pgi_details.csv
├── udise
│   └── sch_safety.zip
│       └── sch_safety.csv
├── udise
│   └── sch_incentives.zip
│       └── sch_incentives.csv
├── pat
│   └── periodic_exam_result_trans.zip
│       └── periodic_exam_result_trans.csv
```

8. Database Structure

This section describes the database structure, the types of database tables and the naming conventions

8.1 Types of database tables

The data tables are divided into static tables and transactional tables. All the data tables can be divided under these four table types:

1. Static Tables
2. Hierarchical Tables
3. Dynamic Tables
4. Aggregated Tables

All the four types of tables are updated in the data dictionary below.

Click on the link below for the data dictionary:
https://docs.google.com/spreadsheets/d/1OM5jClFb3shyk0KKqXk0jtiThcwHntUXdzTyam_lwDo/edit?usp=sharing

Click on the link below for the latest ERD:
https://drive.google.com/file/d/1s1TJUs5c_IZ5zlQvpRWEnb4jWDVMgkI8/view?usp=sharing

9. cQube NIFI data file processing and data validations

This section describes the cQube data file processing calculations and data validations. The files should be loaded into S3 emission bucket as per the emission process schedule timings.

The file processing will be controlled by NIFI and the process will be run as batch mode within the scheduled timing. The NIFI process will be run by the scheduler at 10PM every day as default. The administrator can change the time manually from the NIFI configurations.

All the NIFI process details and the processing time will be captured into the NIFI log files which will be available for the Admin to review.

cQube performance summary

Stage	Type	Zipped File name	Zip file Size	Records	Time taken
Files processed and time taken from API to S3 Output bucket	Static/Master files	school_master	3.2 kb	54728	
		cluster_master	32 kb	3247	
		block_master	1.3 kb	254	
		district_master	1.3 kb	38	
	CRC files	inspection_master	22.4 mb	497075	5 minutes
		user_location_master	57.8 mb	3054698	
	Semester files	semester	128.8 mb	3746754	10 minutes
	Student attendance files(one month)	student_attendance	250.4 mb	10473331	20 minutes
Overall time taken					~35 minutes

9.1 S3 bucket partitioning

S3 buckets will contain partitions for the data files to store. The partitions are created at the S3 input bucket & the S3 Output bucket.

9.1.1 S3 emission bucket partitions

- All the files in the S3 emission bucket will be in the CSV format.
- S3 emission bucket follows the folder hierarchy based on the data sources.
  S3 -> Bucket name -> Data source -> emitted zip files with timestamp

- The folders and the files will be removed from the S3 emission bucket once NIFI copies the data.
- Unprocessed data files will remain in the S3 emission bucket for one week and then they will be deleted automatically at the end of the week.

9.1.2 S3 input bucket partitions

- All the files in the S3 input bucket will be in the CSV format.
- S3 input bucket follows a hierarchical partitioning based on the Data source, Year, Month, date and timestamp
  S3 -> Bucket name -> Data source -> Year -> Year - Month -> date_Source name

  Example for the S3 input bucket:

  ```
  S3/cqube-gj-input/student_attendance/2020/2020-05/2020-05-29_student_attendance
  ```

9.1.3 S3 output bucket partitions

- All the files in the S3 Output bucket will be in the JSON format.
- S3 output bucket follows the hierarchical partitioning based on the data source, Year, Month, date and timestamp, similar to the partitioning that the S3 input bucket follows.
- Metadata files will have information of the latest updated output files which helps cQube to consider the latest output file during the visualization stage.

9.1.4 Data Validation after Ingestion

File size mismatch: NIFI does not allow the file into the S3 Input bucket if the file size does not match. A notification is sent through an email to the data emitter to check the file size and re-emit the data file.

Record count check for the emitted file: Nifi checks if the emitted file records count matches with the original file records count. If the file records do not match, NIFI will not process the file into the S3 Input bucket. A notification email is sent to the data emitter to re-emit the data file.

NIFI fetches the data from the S3 emission data bucket and sends it to the S3 input bucket, after performing the following validations:

- Column level validations: Column datatype mismatch, Number of columns, Data exceeding the column size.
- Improper Data handling: Missing/ null data values for mandatory fields, Empty data files, Special characters, Blank lines in data files.
- Duplicate records validation: NIFI validates the duplicate records by grouping the same kind of records together.

Validation for duplicate records: Record which have duplicate values for all fields (mirror image record) then NIFI will consider the first record and the rest of the records will be eliminated.

Records with duplicate ID: For the rest of the duplicate records where the records are having the same ID (student ID/ assessment ID/ infra ID/ CRC visit ID) and different values will not be inserted into the database tables as ID is the primary key.

Example: For the Duplicate records with different lat long details, NIFI eliminates the records which have the same id and different lat long details or different names or different values.

Records with same values: For semester report, The records which are having the same values for fields Student ID, School ID, semester, studying class and different values for the subjects then NIFI will eliminate those records.

Overlapping data validation: Overlapping data validation takes place based on the data source.

- The NIFI process for student attendance reports will check the last updated day's record from the transactional table and will process the records from the day after the last updated date. The records from all of the previous days will not be considered for NIFI processing
- For the other data sources, duplicate records where the records are having the same ID (student ID/ assessment ID/ infra ID/ CRC visit ID) and different values will not be inserted into the database tables as ID is the primary key.

Other data issues: Data handling in cases like job failures, missing data for certain days and late receipt of the data (receiving data after a few days), updating the wrong data, upon request (when issue identified at the report)

All the validation results will be captured into the NIFI log files as per the table below,

Sl No	Validation type	Error message
1	File size mismatch	"ERROR: file size mismatch for <filename>"
2	Record count check for the emitted file	"ERROR: record count mismatch for <filename>"

3	Validation for duplicate records	“ERROR: duplicate key value violates the data_source” along with record
4	Records with duplicate ID	“ERROR: Detail: Key (school_id)=(<id>) already exists”
5	Records with same values	“ERROR: duplicate key value violates data_source" along with record
6	Overlapping data validation	“ERROR: duplicate key value violates data_source" along with record

Other errors will be displayed with appropriate “ERROR: <error message>”.

Admin dashboard will contain the details of the results of the validations as

1. The last files uploaded with the file name and timestamp
2. Total number of records in the file
3. Number of successfully uploaded records
4. Number of duplicate records
5. Number of invalid records.

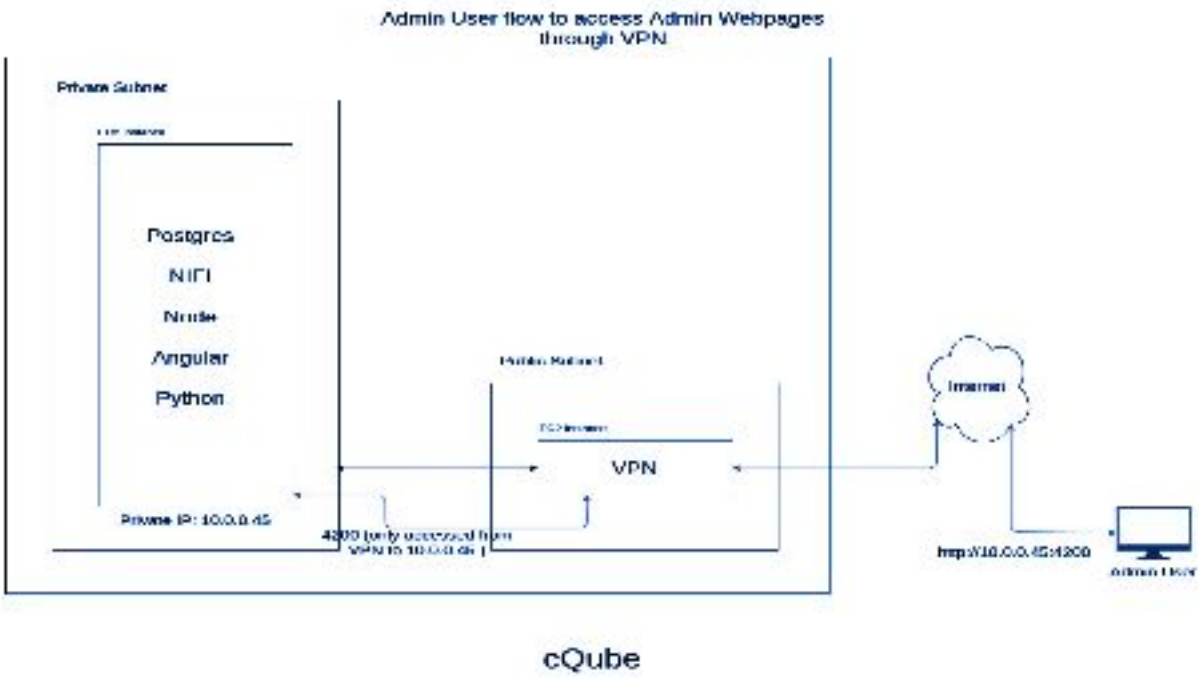
10. Admin login process & Features

cQube has two different interfaces, an interface for the cQube administrator and another interface/ dashboard for the cQube reports:

- Administrator pages: Administrator activities will run separately in the VPN and admin must login through the 2 factor authentication process to perform the admin tasks.Change password functionality will be included in this login.
- cQube dashboard for report: This is a normal login which will have the cQube insights of the metrics.

10.1 Admin login process:

Figure - 5: Admin user flow through VPN



cQube

Admin must follow the 2 factor authentication process to perform the admin tasks. The Admin user should connect to the VPN to avail the 2 factor authentication by following the steps mentioned below:

- Admin has to download the google authenticator app to his phone and register the app with the QR code showing in the OPENVPN Access server page.
- By providing the credentials & Google authenticator code to download the user-locked profile(client.ovpn) file.
- Admin has to install the client openvpn-connect application
- Admin has to create the OpenVPN profile from the client.ovpn file
- Admin has to validate the user authentication and Google authenticator code to login to the cQube VPN.
- With the successful authentication admin can access the cQube admin features by opening the local ip in the browser.

10.1.1 Admin - Features

1. **Create users:**
   - Admin can create new cQube users from the user interface page.
   - Admin will assign the role to the user while creation by choosing the role drop down value.

2. **S3 buckets**

Admin can perform view files and download actions in the S3 buckets by the following steps:

- By using the user interface admin is able to view the raw data files from the S3 input bucket, Metrics files from the S3 output buckets, Transformed and Aggregated data files from the S3 output bucket files.
- Admin can able to download the metrics files in the JSON format, transformed and Aggregated data files in CSV format from the S3 output buckets
- Apart from the user interface Admin users can call the cQube API to download the metrics files in JSON format.
- They can use the same download API to download the transactional data files and aggregated data files in CSV format from the S3 output bucket.
- The admin users use the API to download the raw files from the S3 input bucket in the zip format also.

Admin can enter the AWS console and perform the files cleanup, files Download activities

- Raw data will be downloaded in CSV format and the metrics data will be downloaded in JSON format.

3. **Nifi Scheduler:**

Admin can control the Nifi processes by choosing the start and stop times.

- All the Nifi processes will be listed.
- Schedule Time column is having a drop down box with the hours. By selecting the hour the process will be automatically started at the scheduled time every day.
- The Stopping Hours column is having a drop down box with the numbers. By selecting the number the scheduled running Nifi process will be stopped after the selected number of hours.

4. **View Logs:**

- All the log files will be consolidated into a common folder using soft links.
- New log files will be created every day and the old logs are retained for 7 days. At any point of time, there are 6 log files in the folder, old files are deleted when new files are created.
- Admin will have an option to view the last 200 lines on the browser with the help of the user interface provided. There will be a download option provided to download the entire log file.
- Log files will be downloaded in .log format.

- The list of log files provided to the admin are as follows:
    - Application Logs (Error and Info logs)
    - Admin Logs (Angular and Node logs)
    - System logs
    - NIFI logs
    - Data emission process logs
    - Database logs

5. **Grafana Dashboard Connectivity**

Admin is having an user interface screen "Monitoring Details" screen in the admin dashboard, which connects to Grafana. And also Admin can use the Prometheus APIs to integrate with Grafana for viewing the cQube application health. The following details are displayed in the grafana dashboard:

- RAM usage
- CPU Usage
- Data storage disk usage
- Nifi heap memory usage
- Data source file metrics
- JVM usage

6. **Summary Statistics:**
    - Summary statistics screen is having the processed data information.
    - The records count will be displayed in the Total Records column.
    - The total Number of not processed records are divided into the different validation types such as Blank lines, Duplicate Records, Null records.
    - Total process records count will be displayed in the Processed Records column.
    - Process Start time and Process End time is represented in the remaining columns.

10.2 Report Viewer

- They are defined in a hierarchical manner,
  For example: If the users belong to the State administrative office, they can view the insight statistics of the districts that belong only to that state.
  If the users belong to the district administrative office, they can view the statistics of the blocks which belong to the district.

  The hierarchy is as shown below:

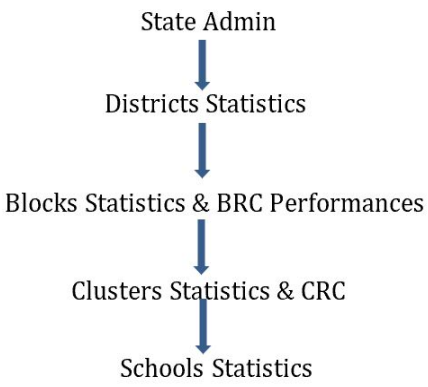


Figure – 6: Report viewer hierarchy

- They can view static dashboards according to their access scope. The selection option dropdown or drilldown options will be provided according to the access scope.
- They will not be able to access any other dashboards which are outside their scope.
- They can have insights into the performance and identify the low and high performers based on the benchmark in the dashboard.
- They can communicate with the users at lower hierarchical levels and provide them with the actionable items or exceptional performance via email (functionality provided in the dashboard)
    - They can alert the low performing district/ block/ cluster/ school based on their performance
    - They can notify the average performing district/ block/ cluster/ school based on their designation, to improve the performance
    - They can recognize the high performing district/ block/ cluster/ school based on their designation, to maintain the performance

11. cQube release upgradation process

cQube upgradation is the updating of the version of the cQube. As of now cQube allows users to upgrade to the latest version from the previous version only.

At the time of a new release users might choose either the installation of the latest version or the upgrade to the latest version.

Steps for upgradation:

- Clone the files from GitHub using the following command:

For example if the cQube upgradation is happening for version 1.3 then the command would be as

$ git clone https://github.com/project-sunbird/cQube/tree/release-1.3

- This downloads the cQube files. The cQube GitHub has all the upgradation files required for upgrading cQube.
- The README.md file has all the instructions that have to be followed for the cQube product upgradation at “Upgradation:” section
- The update.sh script updates the complete cQube stack.
- The update.sh file calls the Ansible playbooks in the background which will complete the cQube upgradation setup.
- The complete upgradation process takes approx 30 minutes.

If the existing users choose the installation of the latest version there is the chance of losing the previous data. To overcome the data lost issue users has to follow the below instructions,

1. Users has to take the previous data backup and keep at some location (either at local machine or store at some S3 location)
2. Data has to be cleaned according to the table changes
   For example:
   - If any column is made as the primary key, the data has to be cleaned of the null values and duplicates.
   - If any columns were removed from the previous tables those column data would be removed.
   - If the total structure of any table was changed then there will not be any provision to restore the data
3. Data should be restored manually once the installation process has been completed.
4. Schedule the nifi jobs after the upgradation is completed.

11.1 Prerequisites

- The latest release branch will be created in Github.
- The latest code of cQube should be committed to the new branch of Github repository
- The existing data structure should be placed as SQLscripts in the new branch of Github repository.
- The latest data structure for cQube should be placed as SQLscripts in the new branch of Github repository.
- All the emission process should be in a stop state, there shouldn’t be any data processing in nifi before the upgradation starts.

11.2 Steps for cQube release Upgradation

- Each release will have to execute an upgrade.sh file.
- upgrade.sh file will call the scripts of ansible.
- Ansible scripts will perform the updates as mentioned below:
- An ansible script will run to take data backup from existing data.
- An ansible script will run the NIFI toolkit. By executing the NIFI toolkit, the NIFI will update the latest flow version to the release environment.
- An ansible script will be run to update the AngularJS code by getting the latest code from Github
- An ansible scripts will be execute the latest SQL scripts by comparing the below database structural changes
    - SQL scripts will check each table from the SQL backup file with the Database schema
    - If the table is already available, All the table columns will be compared.
- If there are no changes in the existing table & table fields, then the table will be skipped and no changes will be made on the existing table.
- If there are any changes in the existing table fields, the Ansible scripts run the alter query to perform the changes on the table
    - If the table does not exist, Then Ansible scripts perform the create the new table using create table query.
- S3 buckets data will not be touched if there are no changes in the existing tables. All the metrics and the calculated data will remain the same.

11.3 Configurations at cQube upgradation process

Configurations are done as a part of the installation process by the ansible scripts. All the properties of the sensible information like URLs and passwords are saved in the ansible properties file which is encrypted by default. Configuration is an automated process, no manual interactions are required in this stage.

Enable / Disable the process groups:

This configuration should be performed at the installation / Up-gradation stage of cQube. The cQube user may select the process groups which they wanted to include into the cQube. While the installation / Up-gradation of the cQube user will be available to select the existing process group

by giving the true / false in the datasource_config.yml document which is available at the below link.

https://github.com/project-sunbird/cQube/blob/release-1.3/ansible/installation_scripts/datasource_config.yml

All the configuration should be given like below,

nifi_telemetry: false
nifi_crc: true
nifi_attendance: true
nifi_semester: true
nifi_infra: true
nifi_diksha: true
nifi_udise: true

As of now there were a total of 7 processor groups available in cQube.

11.4 Limitations

- Metrics will be regenerated if there are any changes in the existing table fields such as adding/deleting the new columns to the existing tables.
- Metrics will be regenerated if there are any structural changes at the data source.
- The output files should be regenerated with the recalculated metrics by having the modified fields information.

12. Reports

As earlier said cQube is a “decision making tool”.

In explanation, cQube’s every report in Analytics is made up of dimensions and metrics. cQube reports provide good predictive insight anticipates upcoming correlations at different components of the data and activities of the educational system. These reports help cQube admins to take an appropriate decisions based on the insights.

For the visualization of the metrics cQube uses the below types of reports.

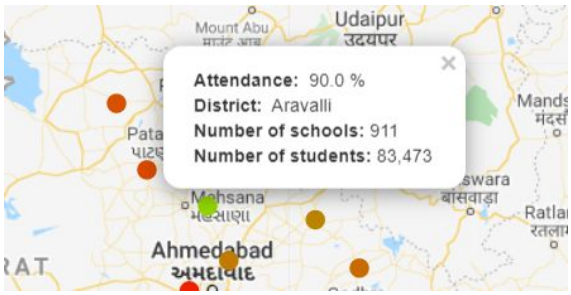
1. Map based visualization
2. Scatter plot visualization
3. Stacked bar report
4. Table format visualization
5. Bar chart visualization

More details of the visualization feature wise details are explained below,

12.1 Map based visualization

Landing page of the Map report will be represented for the whole state by having the colour gradient dots on the different districts of the state. Colour gradient will be declared from Red-Green where Red represents the poor performance and the Green represents the good performance.

Tooltip: On mouse hover of the dot, The user will be able to see the tooltip which can give the results in formation of the metric calculations. Along with the results the tooltips are containing the data like id and the location name and location details also.



Blocks button: Block button will be a clickable function and it will display the information and the metrics results with the color gradient dots, tooltip information of the total blocks of the state.

Clusters button: Clusters button will be a clickable function and it will display the information and the metrics results with the color gradient dots, tooltip information of the total clusters of the state.

Schools button: Schools button will be a clickable function and it will display the information and the metrics results with the color gradient dots, tooltip information of the total Schools of the state.

Drill down functionality: User will be redirected to the next hierarchical level of the locations by clicking on the dots.
For example: If the user clicks on some district's dot, then the visualization will be populated by all the blocks of the districts.
If the user clicks on some block's dot, then the visualization will be populated by all the clusters of the districts.

Drop down functionality: Drop down functionality will be experienced by the selection from the "Choose a District", "Choose a Block" and "Choose a cluster" drop down box values.

Time series: By selection of the time represented dropdown values users will get the time series information. Time series will be changed according to the data sources.

- Student attendance report has the Year and Month time series selection.
- Semester report is having the Semester wise time series selection.

- Infrastructure report is not having the time series, It represents the whole information of the infrastructure data.
- UDISE is having the Previous day, last 7 days & 30 days time selection.

Download Functionality: The download icon provides the download functionality of the report metrics based on the selection of the drop down values.

12.2 Scatter plot visualization

The landing page of the scatter plot report will be represented for the whole state by having the several normalised metrics using the raw data and you can read more about them.

A key feature of this dashboard is its ability to Zoom In and Out at various administrative levels. The administrative levels include District, Block and Cluster.

This has been done to provide relevant insights at the appropriate administrative level. In addition to visualising data, the dashboard also gives you the ability to download the data at various administrative levels.

This feature has been enabled to provide freedom to power users to derive additional insights that may not be captured in this dashboard. You can download the data using the dropdown option on the top right corner.

Table representation: The table in the scatter plot will have the different insights as columns of the table which represent the performances of different educational areas.

X-Axis & Y-Axis: these are the dropdown boxes which contain the columns of the above mentioned table. By selecting the x-axis value and the y-axis value the dot will be displayed at the cross point of the x & y results.

Drop down functionality: Drop down functionality will be experienced by the selection from the “Choose a District”, “Choose a Block” and “Choose a cluster” drop down box values.

Download Functionality: The download icon provides the download functionality of the report metrics based on the selection of the drop down values.

12.3 Stacked bar report

The landing page of the scatter plot report will be represented for the whole state by having the several normalised metrics in the bars with the combination of multiple items. The results will be represented in the x & y axis representation.
Example: Different subjects will be represented in a single bar between the X-Axis & Y-Axis.

Drop down functionality: Drop down functionality will be experienced by the selection from the “Choose a District” drop down box values.

Time series: By selection of the time represented dropdown values users will get the time series information. Time series will be changed according to the data sources. Diksha is having the Previous day, last 7 days & 30 days time selection.

Download Functionality: The download icon provides the download functionality of the report metrics based on the selection of the drop down values.

12.4 Table format visualization

Diksha is using the table format representation. The landing page of the table format visualization is having the metrics of in combination with the ‘Textbook content plays’ and the ‘Course content plays’

Content Selection: Content selection is the dropdown box which will represent the overall content, Textbook content and course content. By selecting the value table will be filtered as per the selected value.

Drop down functionality: Drop down functionality will be experienced by the selection from the “Choose a District” drop down box values.

Time series: By selection of the time represented dropdown values users will get the time series information. Time series will be changed according to the data sources. Diksha is having the Previous day, last 7 days & 30 days time selection.

Download Functionality: The download icon provides the download functionality of the report metrics based on the selection of the drop down values.

12.5 Bar Chart visualization

Diksha is using the bar chart visualization. The landing page of the bar chart visualization is having the metrics of in combination with the ‘Textbook content plays’ and the ‘Course content plays’ in the form of horizontal bars.

Content Selection: Content selection is the dropdown box which will represent the overall content, Textbook content and course content. By selecting the value table will be filtered as per the selected value.

Choose collection Name: By choosing the Collection name from the dropdown the bars will represent the selected results.

Drop down functionality: Drop down functionality will be experienced by the selection from the “Choose a District” drop down box values.

Time series: By selection of the time represented dropdown values users will get the time series information. Time series will be changed according to the data sources. Diksha is having the Previous day, last 7 days & 30 days time selection.

Download Functionality: The download icon provides the download functionality of the report metrics based on the selection of the drop down values.

Data source wise reports Visualizations are as below

Student Attendance Report: Map visualization
Semester Report: Map visualization
CRC Report: Scatter plot visualization
School Infrastructure Reports: Map Visualization and Scatter plot visualization
Diksha: Stacked bar, Bar chart and table type visualizations
UDISE: Map visualization
PAT: Map visualization
Composite Report : Scatter plot visualization
Telemetry : Map visualization