# Deployment blueprint of Audit and Enterprise Search Stack using docker compose

Deployment of the complete environment is facilitated by Docker Compose and it deploys postgres database, mobius  server and view containers, Kafka and event analytics service.

**Prerequisites**:

* Windows with 4 vcpu and 32 GB RAM or Linux VM with 4 vcpu and 16GB RAM
* Docker desktop installed on windows or docker engine  and docker-compose installed on Linux.
* Internet access to download docker images from public registry, download docker tools and Mobius Local User (non-production) images from Access Portal
* Browser Chrome or Edge.

**Installation using docker compose:**

This section describes the step-by-step installation of the complete environment or docker stack.

1. Download mobius-server, mobius-view, and eventanalytics docker images from the public registry registry.asg.com.

docker login registry.asg.com

docker pull registry.asg.com/mobius-server:12.0.0

docker pull registry.asg.com/mobius-view:12.0.0

docker pull registry.asg.com/eventanalytics:1.1.0

2.  (OPTIONAL) If you don't have access to public registry, but have received zip file containing exported docker images, then use the following to import all the images to local docker registry.

docker load -i mobius-docker-image.tar

docker load -i mobius-view-docker-image.tar

docker load -i eventanalytics-docker-image.tar

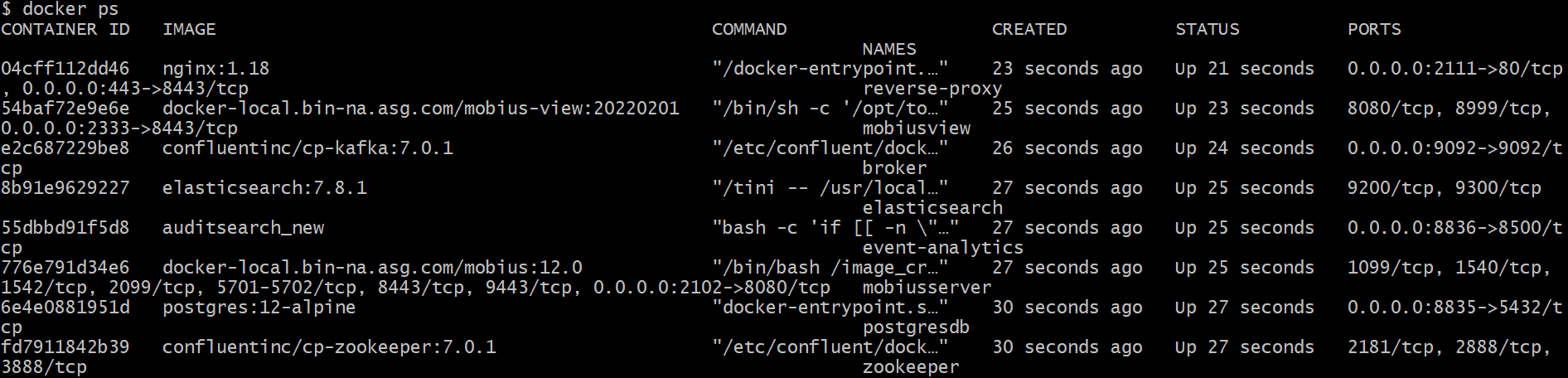
3.  Unzip mobius-basic\*.zip file and navigate to the unzipped directory.

* 1. Update .env file:
     1. Update reference to mobius server, mobius view and event analytics images if needed ( optional)
     2. Update mobius view license key

4.  Run the following command from the unzipped directory to start Mobius deployment. "-p" command  will associate deployment configuration with 'mobius12beta' project. All volume and networks will be created with "mobius12beta" prefix. All docker-compose command must run with same project name for this deployment. Recommendation is to use all lower case for custom project name if you don't want to use 'mobius12beta' for your project.

docker-compose -p mobius12beta2 up -d

5.  Command "docker ps" will show the following output.



**Turn on Auditing using mobius vdr admin**

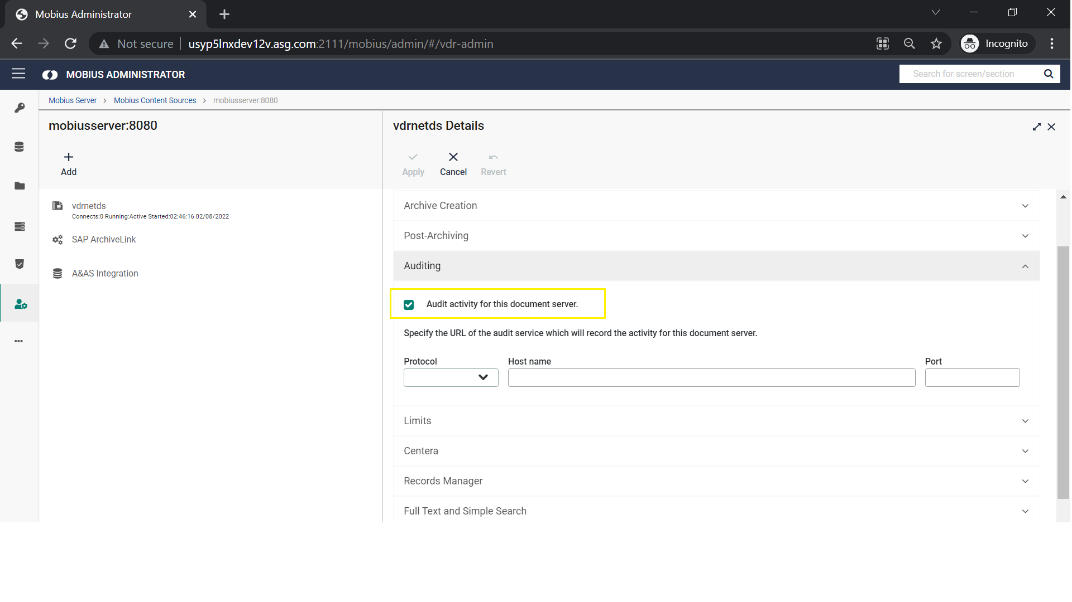
Mobius has to be configured manually for turning on auditing. Perform the following configuration steps:

1) Login to http://<hostname>:2111/mobius/admin/

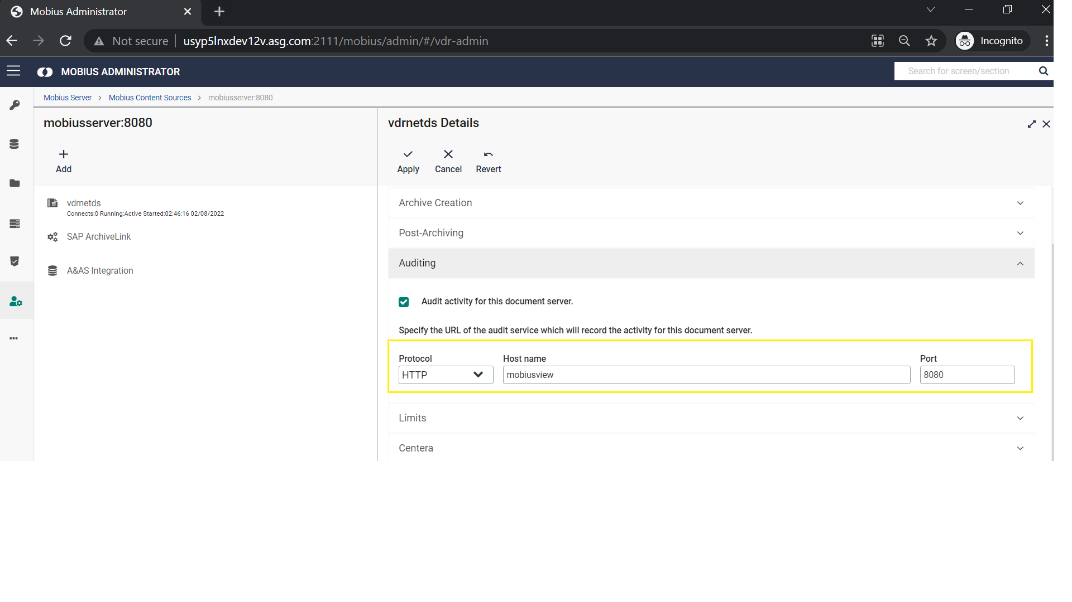
2) Go to VDR Admin i.e. http://<hostname>:2111/mobius/admin/#/vdr-admin

3) Go to document server and open the document server details/properties

4)  Enable Audit activity for this document server under **Auditing**

****

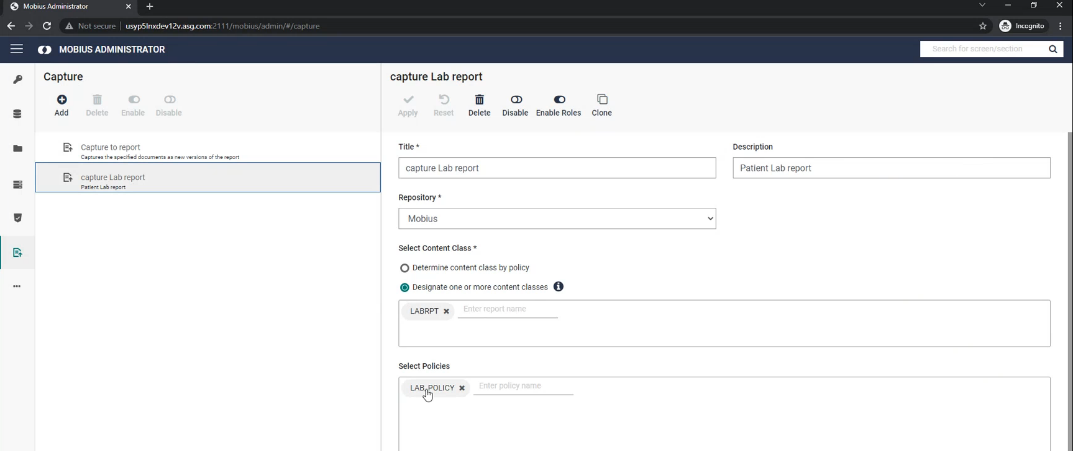
On the same panel, select Protocol as http and provide Host name as mobiusview & Port as 8080 (note that the Host name & Port are as specified for the mobiusview container in the docker-compose.yml file)



Click Apply and save the configuration.

5) You should create capture target with existing policy or by adding a new policy.

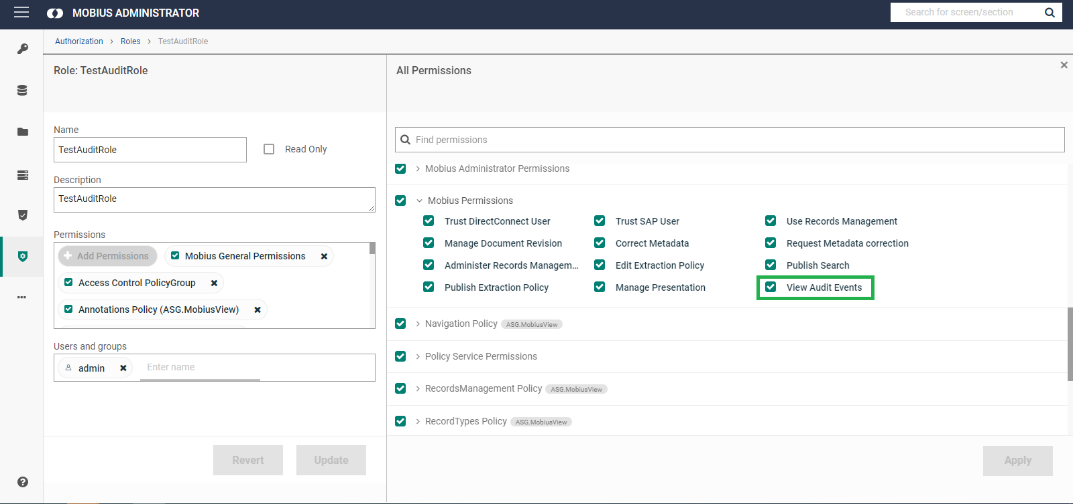
6) Create a capture target



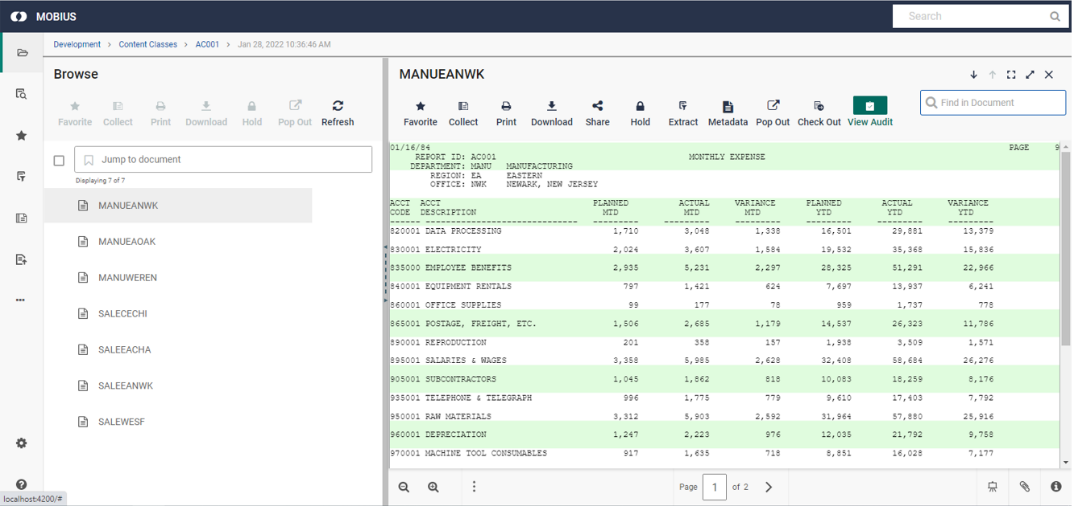
Capturing above reports or any other browse interactions with report and topics will send audit event data to kafka which are stored in postgres database.

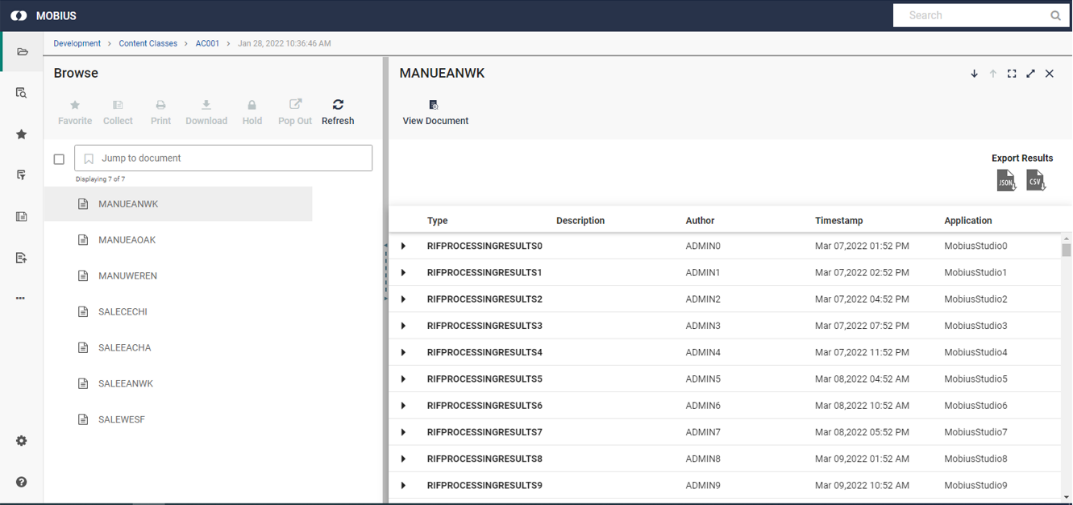
### Enabling Audit access to users in Mobius View

In Mobius Admin, under Authorization context, user's role must have 'View Audit Events' permission checked.



Once the user has 'View Audit Events' permission, he can see 'View Audit' option in document viewer of Mobius View.





**Adding Index (Topic) information to your event reporting database**

Event database only contains audit events data by default in reporting database. This sections describes a way to import mobius index data to event database so that you can analyze event data along with mobius index data,

This section will show you how it's possible to export any mobius index information associated with your documents in Mobius to your event reporting database, and use that information along with the events to answer questions about what events are associated with content in Mobius, e.g. what events occurred for an index SSN with a value of 222-333-5678 in the last year? Mobius has additional information about the documents which you can also use to analyze your event stream, in the form of indexes. Here we'll explain how to export that information in batch to your event reporting database, to let you do analysis including the index information.  These export operations happen for any set of documents you specify for the export, and you can run these type of operations periodically if you want to export new information, or update changes to existing indices in the Mobius database.

The index information can be exported from your Mobius installation to your reporting database via batch operations, using the MobiusRemoteCLI utility, MobiusRemoteCLI is part of non-production mobius local user package.

You'll have to configure your MobiusRemoteCLI if you have not already; an example configuration for the repository needs to be set, as well as configuring the CLI for basic authentication. Here's an example of configuring the MobiusRemoteCLI to talk to the MobiusView instance deployed with your docker-compose.yml:

> MobiusRemoteCLI.bat configure

Mobius CLI tool

Copyright (C) 2020-2022, ASG Technologies Group, Inc. All rights reserved.

List of Configurations Available:

[1] Repository

[2] Basic Authentication

[3] Oauth Authentication

[4] OAuth Client Credentials

[5] Headers

[6] Cookies

[\*] Quit

Please enter one of the [] values: 1

List of Actions Available:

[A] Add

[U] Update

[L] List

[R] Remove

[#] Previous Menu

[\*] Quit

Please enter one of the [] values: A

Repository

Repository Name: Mobius

Repository url: http://localhost:2111/mobius/

Repository User Name: ADMIN

Repository Password:

Save or Cancel [Y, N]: Y

The configuration has been saved successfully.

List of Actions Available:

[A] Add

[U] Update

[L] List

[R] Remove

[#] Previous Menu

[\*] Quit

Please enter one of the [] values: #

List of Configurations Available:

[1] Repository

[2] Basic Authentication

[3] Oauth Authentication

[4] OAuth Client Credentials

[5] Headers

[6] Cookies

[\*] Quit

Please enter one of the [] values: 2

List of Actions Available:

[S] Set

[V] View

[U] Unset

[#] Previous Menu

[\*] Quit

Please enter one of the [] values: S

User Name [admin]: admin

Password [<<Hidden>>]: admin

Save or Cancel [Y, N]: Y

The configuration has been updated successfully.

List of Actions Available:

[S] Set

[V] View

[U] Unset

[#] Previous Menu

[\*] Quit

Please enter one of the [] values: \*

Exporting Indices from Mobius Server

In order to tell Mobius server to export information to the reporting database, we can submit a job using the MobiusRemoteCLI ftijobsubmit command. This command has options which allow you to select any set of Content classes, or particular content ingestions/archives using a variety of options described below:

> MobiusRemoteCLI.sh ftijobsubmit -h

Mobius CLI tool

Copyright (C) 2020-2022, ASG Technologies Group, Inc. All rights reserved.

Usage: ftijobsubmit [options]

Options:

-help, -h, ?, --help

Print usage

\* -s, --doc-server

Content repository name

-u, --user

Recipient ID

-p

Password to the repository

-e, --error-level

Error level. Value must be one of {i,w,x,e,d,I,W,X,E,D}

Default: x

-o, --output-file

Output filename

\* -r, --report

Content class ID. Content class IDs support \* and ? wildcards. Surround

wild-carded content class ID specifications with double quotes (e.g.,

"ABC\*") to avoid shell expansion

-v, --versions, --version

ingestion IDs support \* and ? wildcards when a single ingestion ID

is specified. Surround wild-carded ingestion IDs with double quotes

(e.g., "201801\*") to avoid shell expansion

-vr, --version-range

Ingestion-ID-range-start Ingestion-ID-range-end

--verbose

Display more information

Default: false

-t, --jobtype

{MERGE, UPDATE, ADD, TOPIC\_EXPORT}

Default: MERGE

-i, --ignoreErrors

Job processing will not be halted due to error processing individual

versions

Default: false

Following example export all indexes for content class: LABRPT:

> MobiusRemoteCLI.sh ftijobsubmit -s Mobius -r LABRPT -t TOPIC\_EXPORT

You can run this operation multiple times, and it will replace any existing information with the current state of the Mobius database for the selected export operation. If index change in Mobius, rerunning this command for the affected documents will overwrite the up to date information in the content database. Additionally, you can submit jobs for ranges of versions, or wildcarded sets of versions, to choose what topic information to export.

At this point, the index information for any LABRPT documents will be in the reporting database.

The data will be available in tables in the reporting database called documents, in json format, and is merged with event data in a view called documentevents, which can be queried directly or used as the basis for graphs in Superset or reporting tools. Next section explains how can you deploy and configure Apache superset to analyze events data.

# Apache Superset

Apache Superset is an opensource data visualization tool that can be used to access auditing data captured by the Rocket Audit and Enterprise Search Service.  The following section describes how to deploy Apache Superset as a Docker image.  Information relating to the Docker image can be found in docker hub at <https://hub.docker.com/r/apache/superset>.  Further information relating to Apache Superset can be found at <https://superset.apache.org/>.

## Deploy Superset

Superset Docker image can be deployed using the following docker commands.  It can be deployed on the same machine as the Audit and Enterprise Search deployment or another as long as they are on the same network and are visible to each other.

### Start Superset Container

Run the superset container with the following command.

docker run -d -p XXXX:8088 --name superset apache/superset

where XXXX is an available port on the docker host machine.

Note : All the below commands listed below assumes the name of the container is superset as defined in above example

### Setup Superset Admin Account

Run the following docker commands to create the Superset admin account.

docker exec -it superset superset fab create-admin --username admin --firstname Superset --lastname Admin --email admin@superset.com --password admin

### Upgrade Superset Database to Latest Version

Perform the following command to ensure Supersets Database is using the latest version by executing the following command.

docker exec -it superset superset db upgrade

### Load Examples (OPTIONAL)

Superset can be configured with a number of example charts and dashboards, that can be used for learning purposes.  The following command will configure Superset to install and configure this data.  This data is not needed for working with Audit and Enterprise Service data.

docker exec -it superset superset load\_examples

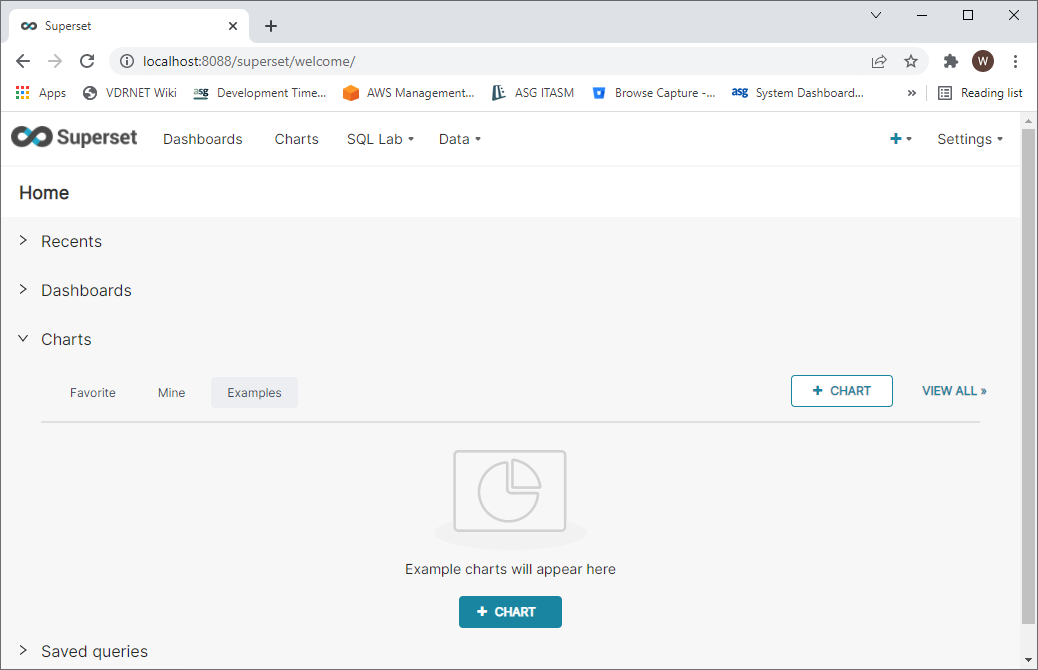
### Setup Roles

Superset is role based the following command will create default roles.

docker exec -it superset superset init

### Login to Superset

Using a browser navigate to the Superset server http://<superset-host>:<superset-port>/login, use the credentials that you setup above in Setup Superset Admin Account. (admin/admin)

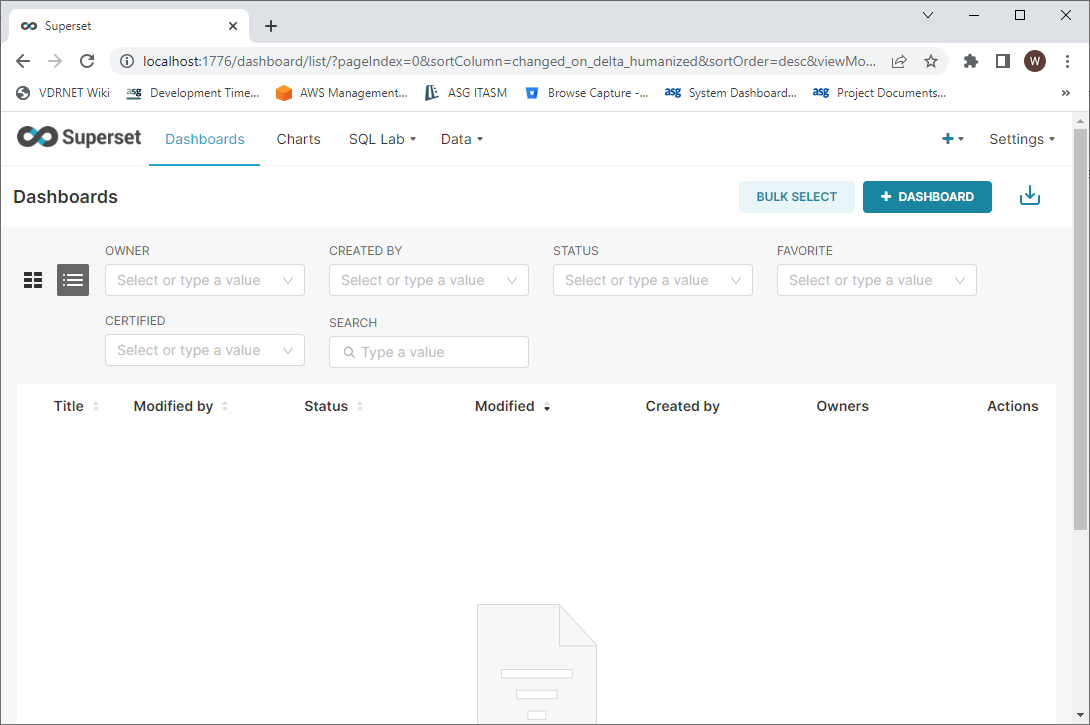


## Import Demo Dashboard (Optional)

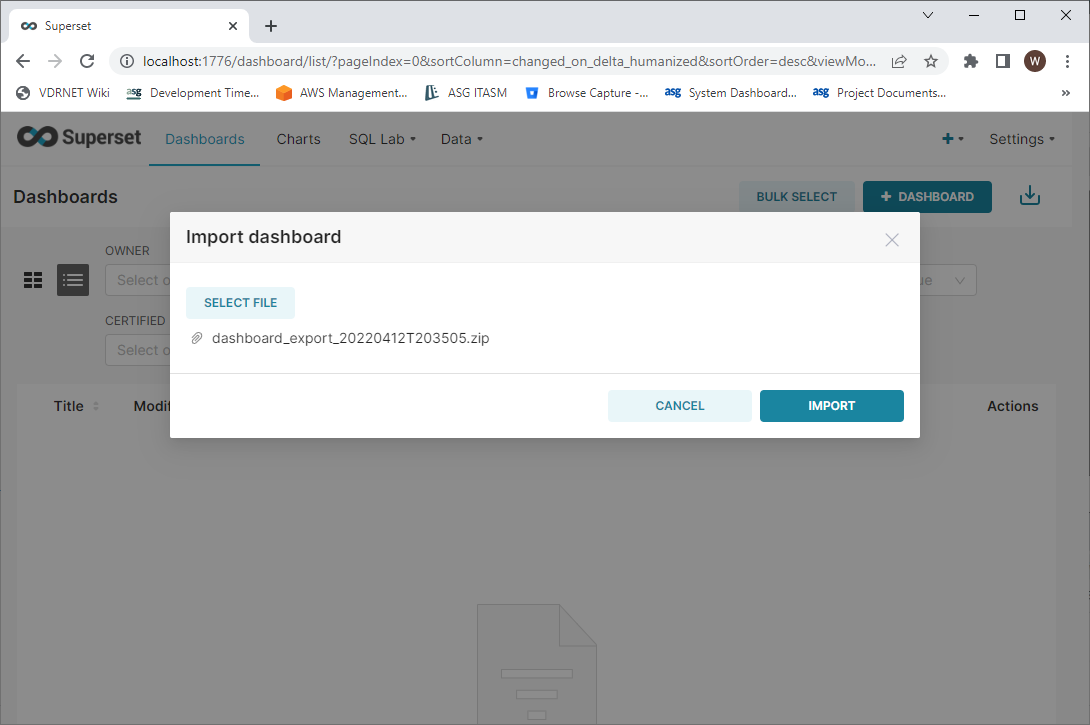
**Requires that sample data be restored from above**

The following will import a demo dashboard into Superset to be used with the Audit service.  This dashboard uses the sample data provided in the zip file which must be restored using the optional restoration process defined above.

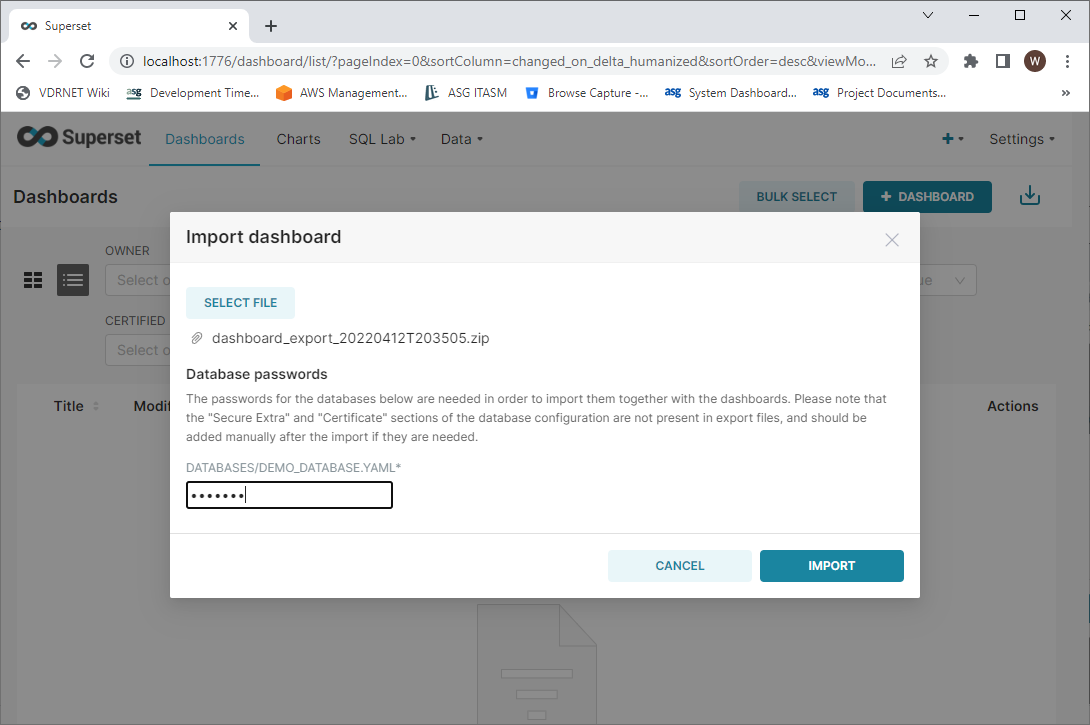
* In the non-production zip file there will be a exported dashboard file named dashboard\_export\_20220412T203505.zip.  This file will be imported to the Superset installation.
* In the Superset GUI click on the Dashboards menu item to go to the dashboards page.
* To the right of the +DASHBOARD button click on the Import Dashboard icon.



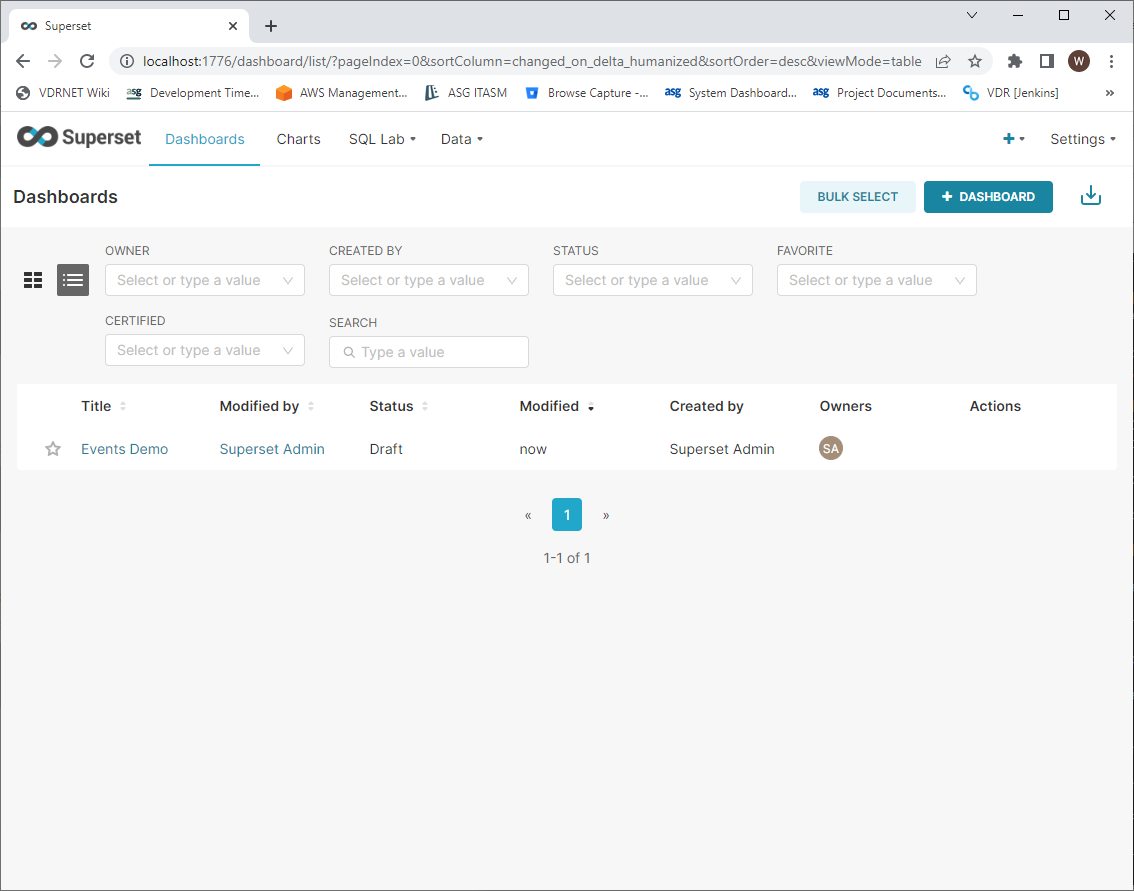
* Click on the SELECT FILE button nd locate the dashboard\_export\_20220412T203505.zip dashboard export file to be imported and click on IMPORT.



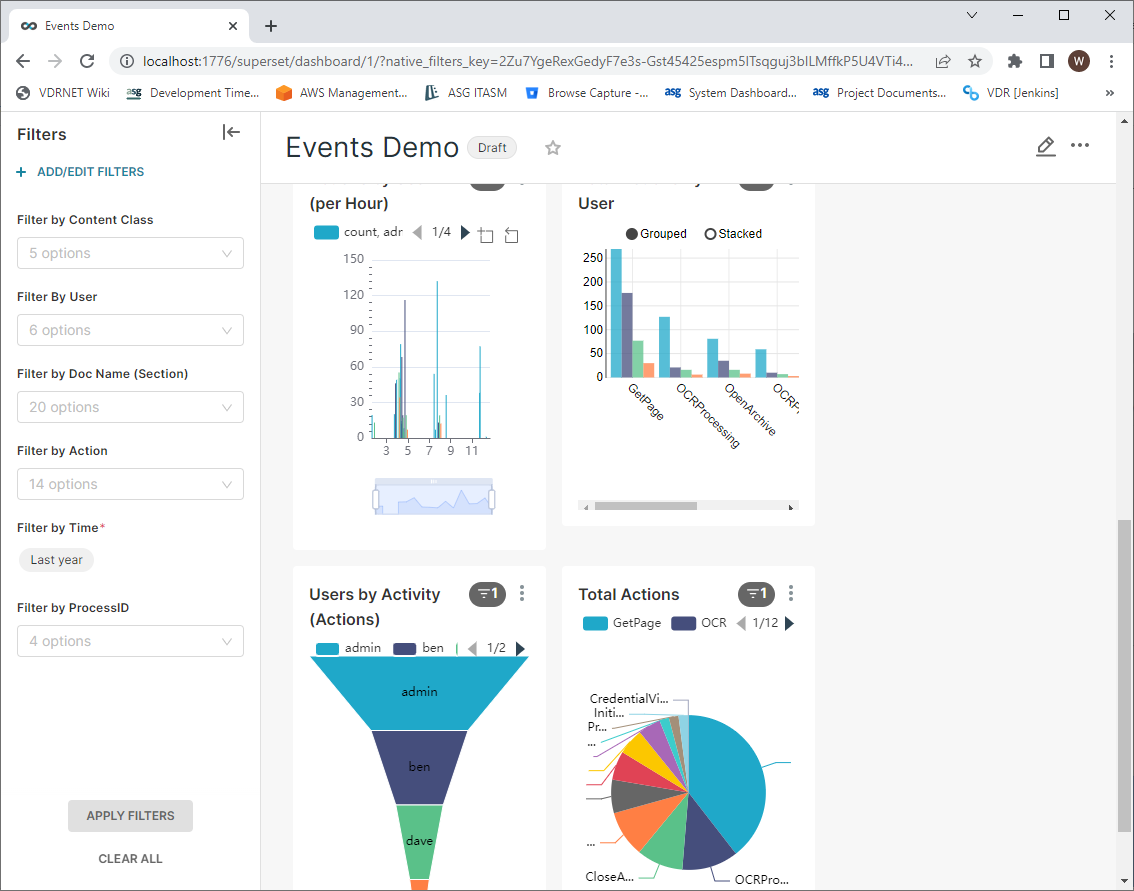
* Provide the database password for the dashboard, it will be "postgres" in this case.  Click on IMPORT.



* The newly imported dashboard should now appear in the list of dashboards



* Click on the events dashboard to reveal the charts it contains.

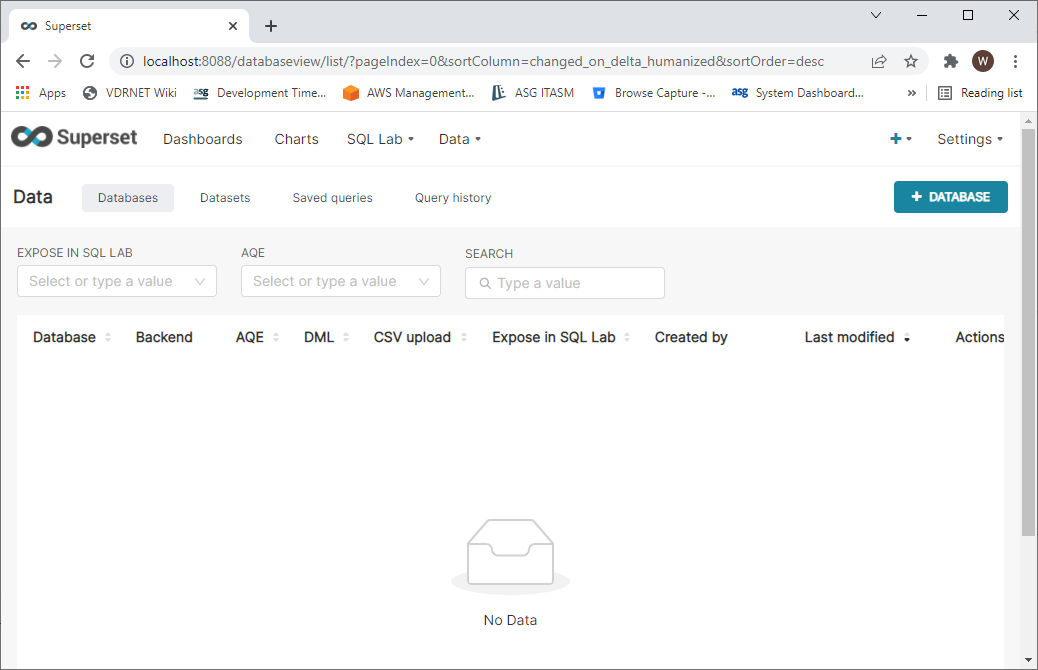


## Configure Superset to Connect to Audit and Enterprise Search Database

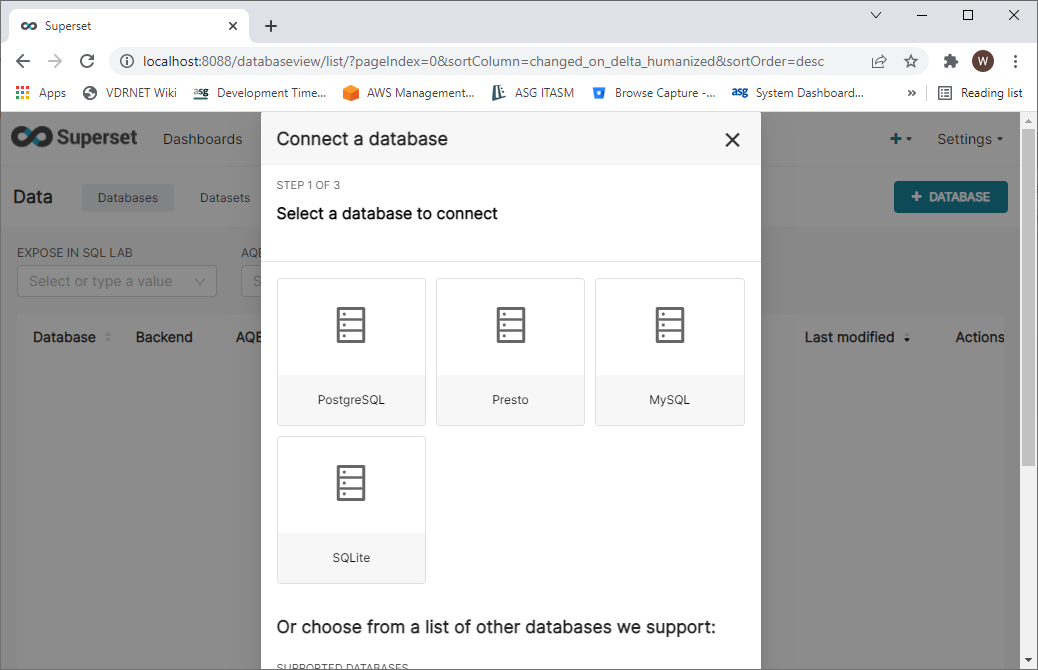
**If you have not imported the Sample Dashboard follow these steps to connect Apache Superset to the Events database or you want to connect to a different deployment of Audit follow these steps to connect.**

In order for Superset to Visualize data from the Audit and Enterprise Search Database a Superset database connection must be established.

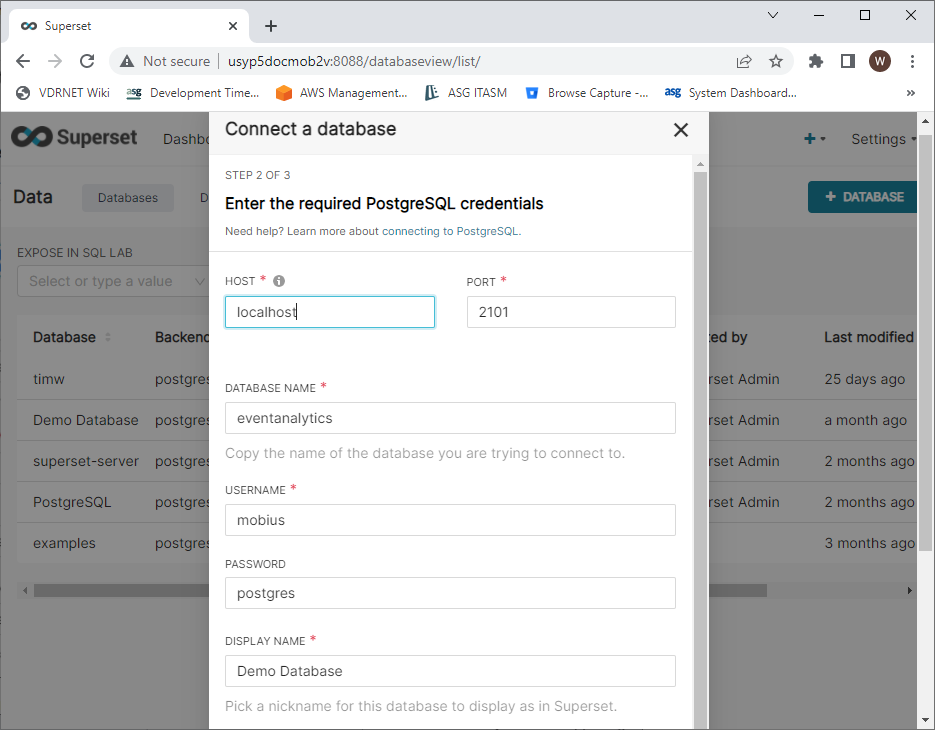
* In Superset GUI navigate to the Data|Databases menu item to display the databases UI.

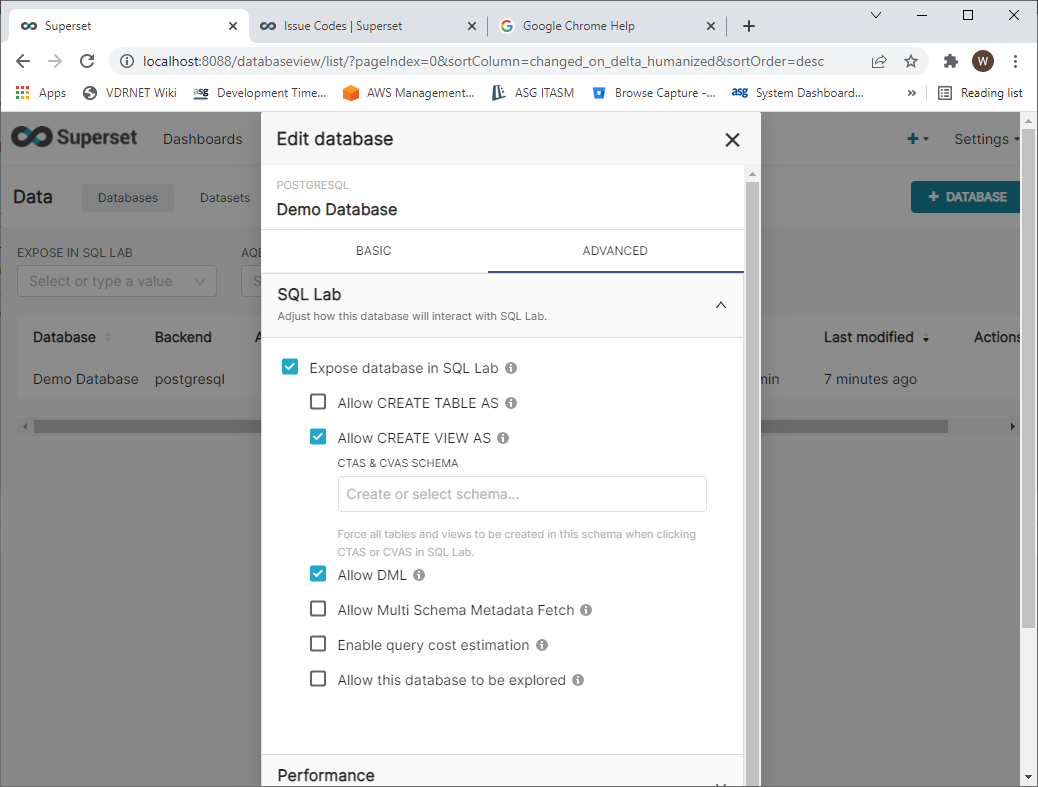


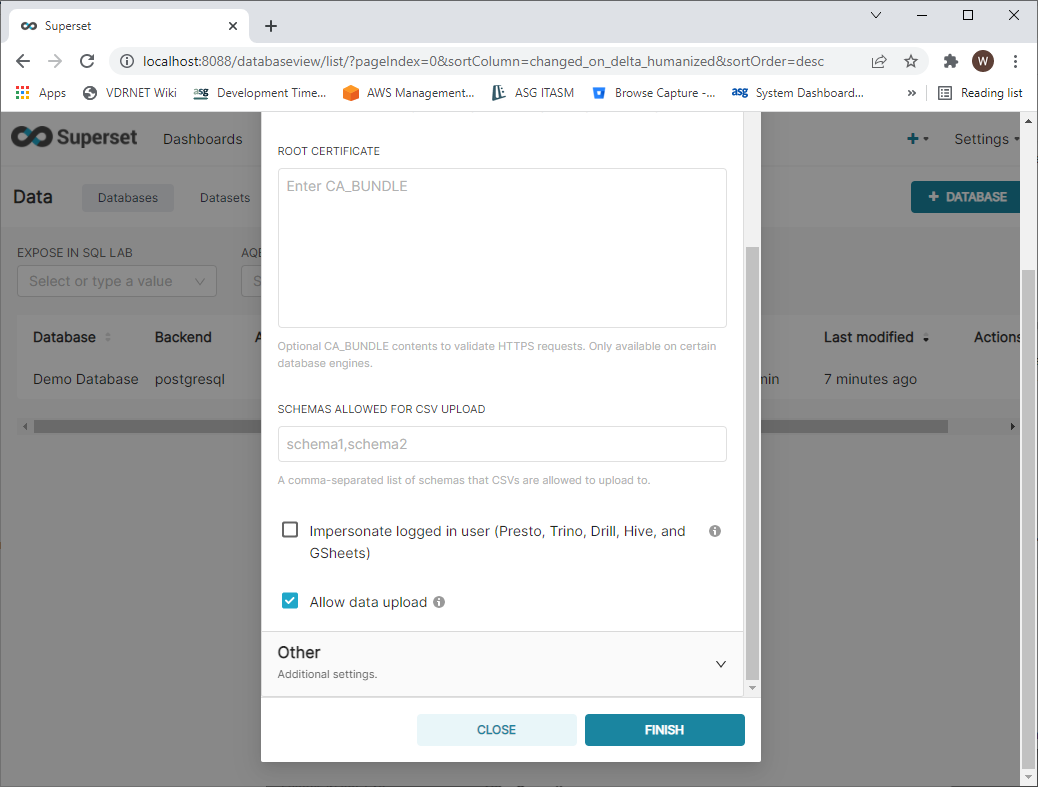
* Click on  the +Database button to create a new database connection.



* Select the PostgreSQL icon to create a new PostgreSQL connection



* Fill in the hostname with the hostname of the deployed Audit Service, PORT: **2101**, Database: **eventanalytics**, Username: **mobius**, Password: **postgres**, Display Name: **Demo Database** and click on Connect button.  The dashboard import step listed below assumes the display name as  **Demo Database** as defined in this step.  **If you have previously imported the Example Dashboard, then choose a different display name than "Demo Database"  since it will have already been created.**  
    
  
* Click on the SQL Lab tab and check Expose database in SQL Lab, Allow Create VIEW AS and Allow DML check boxes.

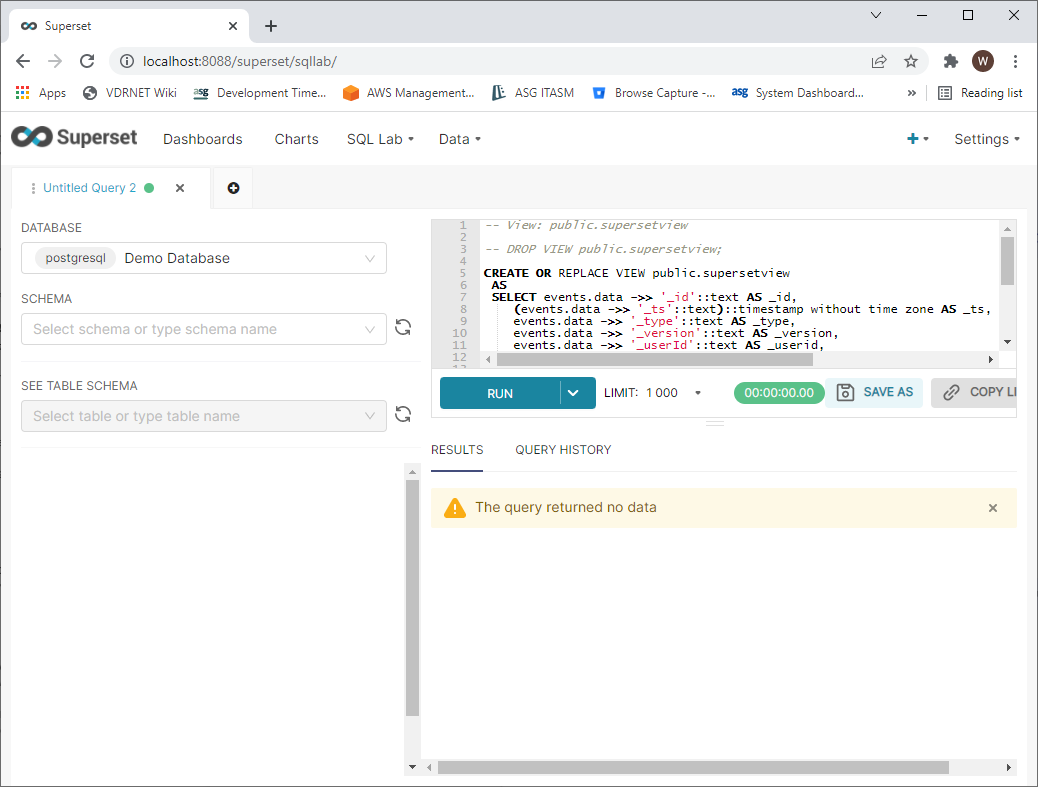


Click on the security section and check the Allow Data Upload check box and click Finish Button.

## Create View to Access Data

In order to work more conveniently in Superset, a view will be created to map JSONB field data items to regular relational  columns.  These columns will be referenced in charts instead of the jsonb field format. If you have done a backup / restore of data using scripts, then this step is optional. You can proceed to "Import Demo Dashboard" step.  **If you have already restored the sample data above, then this step is not necessary unless you are connecting to a new deployment of Audit service.**

* In the Superset GUI, click on the SQL Lab|SQL Editor menu item to reveal the SQL Editor.



## Select the Audit database from the Database drop down and copy and paste the following SQL script to the SQL editor window and click run to create the view.

-- View: public.supersetbetaview

-- DROP VIEW public.supersetbetaview;

CREATE OR REPLACE VIEW public.supersetbetaview

AS

SELECT events.data ->> 'id'::text AS \_id,

(events.data ->> 'ts'::text)::timestamp without time zone AS \_ts,

events.data ->> 'type'::text AS \_type,

events.data ->> 'userId'::text AS \_userid,

events.data ->> 'action'::text AS \_action,

events.data ->> 'description'::text AS \_description,

((events.data ->> 'mobius'::text)::json) ->> 'server'::text AS \_mobius\_server,

((events.data ->> 'mobius'::text)::json) ->> 'recipient'::text AS \_mobius\_recipient,

((events.data ->> 'mobius'::text)::json) ->> 'section'::text AS \_mobius\_section,

((events.data ->> 'mobius'::text)::json) ->> 'contentClass'::text AS \_mobius\_contentclass,

((events.data ->> 'mobius'::text)::json) ->> 'contentIngestion'::text AS \_mobius\_contentingestion

FROM events;

ALTER TABLE public.supersetbetaview

OWNER TO mobius;

-- View: public.supersetbetadocumentview

-- DROP VIEW public.supersetbetadocumentview;

CREATE OR REPLACE VIEW public.supersetbetadocumentview

AS

SELECT documentevents.data ->> 'id'::text AS \_id,

(documentevents.data ->> 'ts'::text)::timestamp without time zone AS \_ts,

documentevents.data ->> 'type'::text AS \_type,

documentevents.data ->> 'userId'::text AS \_userid,

documentevents.data ->> 'action'::text AS \_action,

documentevents.data ->> 'description'::text AS \_description,

((documentevents.data ->> 'mobius'::text)::json) ->> 'server'::text AS \_mobius\_server,

((documentevents.data ->> 'mobius'::text)::json) ->> 'recipient'::text AS \_mobius\_recipient,

((documentevents.data ->> 'mobius'::text)::json) ->> 'section'::text AS \_mobius\_section,

((documentevents.data ->> 'mobius'::text)::json) ->> 'contentClass'::text AS \_mobius\_contentclass,

((documentevents.data ->> 'mobius'::text)::json) ->> 'contentIngestion'::text AS \_mobius\_contentingestion,

documentevents.documentdata ->> 'page\_count'::text AS pgcount,

documentevents.documentdata ->> 'page\_of\_doc'::text AS docpage,

documentevents.documentdata ->> 'section\_id'::text AS docname,

documentevents.documentdata ->> 'ProcessID'::text AS processid

FROM documentevents;

ALTER TABLE public.supersetbetadocumentview

OWNER TO mobius;