

Tutorial: Collaborative Data Science with Interactive Workstations and Databases

Contents

Introduction	5
Learning Objectives	6
Build Data Analysis Workstation	8
Run the pfda-ttyd Featured App	8
Present a simple web server on port 8080	9
Deploy PostgreSQL, psql, pgadmin, and RStudio	10
Access the postgres-local DB from pgadmin	11
Create a new database and tables	12
Load the cluster database from delimited text files	12
Copy the data into the cluster DB tables	13
Share Files Between Workstation FS and PostgreSQL Docker FS	14
Access the postgres-local DB from rstudio	14
Share Files Between Workstation FS and RStudio Docker FS	16
PostgreSQL Tips	16
Force drop connections.....	16
Quick queries for estimated row counts.....	16
Build SAS Studio Workstation	17
SAS Studio Workstation Space	17
Run the pfda-ttyd Featured App with SAS Studio Snapshot	17
Open an SDTM File in SAS Studio	19
Build KNIME Workstation	20
Run the guacamole Featured App	20
Install Additional Utilities and Dependencies	23
Install and Start KNIME	23
Install US City Geo Data Using the Chromium Browser	25
Deploy Local PostgreSQL DB Server and CLI	26
Deploy pgadmin and Connect to the Local DB	27
Share Files Between Workstation FS and PostgreSQL Docker FS	28
Add Shell and SQL Scripts for Use With KNIME	28
Download the KNIME Workflow	29
Run the KNIME Data Transformation Workflow	29
Import and Open the Workflow and Update Dependencies.....	29
Set the pFDA CLI Auth Token and Data Folder Variables.....	31

Create the DB.....	32
Download the Data from precisionFDA Folder.....	32
ETL the Data into the DB.....	32
Analyze the Data and Create Reports.....	33
Publish the Reports to precisionFDA My Home.....	34
Deploy a precisionFDA Database Cluster	35
Create the Database	35
Connect to the cluster DB from pgadmin	36
Create a new database and tables	37
Load the cluster database from delimited text files	38
Create and upload delimited data files.....	38
Create and upload a manifest of data file IDs.....	39
Download the files in the manifest to the Data Analysis Workstation.....	40
Iterate through manifest and download data files	40
Copy the data into the cluster DB tables	41
Connect to the workstations_and_databases_tutorial_db cluster database.....	41
Copy the patients and observations data into the cluster DB.....	41
Connect RStudio to the cluster DB	42
Backup the cluster DB and restore it to local DBs	43
Add a postgres role to the cluster DB	43
Backup the cluster DB using pgadmin	44
Copy the backup file from the pgadmin container to the workstation filesystem	45
Upload the backup file to precisionFDA	46
Restore the backup to the data analysis workstation local DB	47
Restore the backup to the data analysis notebook local DB	49
Stop or Terminate the Database Cluster	51
Build Data Analysis Notebook	52
Run the pfda-jupyterLab Featured App	52
Download and Install the pfda CLI	54
Deploy Local PostgreSQL DB Server	55
Create a Table with some data in the Local DB	56
Create a Notebook and Connect to the Local DB	57
Connect to the Cluster DB	57
Load a Complete Notebook from a Snapshot	58

Build Epidemiology Data Analysis Notebooks	60
Copy the Notebook Resources to a New Private Space	60
Running in non-interactive mode with papermill	61
Build Epidemic Modeling Notebook	61
Build Time Series and Outbreak Detection Notebook	63
Build Contact Tracing Notebook	63
Snapshot and Terminate Workstations	65
Stop the Docker Containers and Snapshot Data Analysis Workstation	65
Terminate the Workstation.....	65
Snapshot and Terminate the Data Analysis Notebook	65
Terminate the Workstation and Notebook and Database Cluster.....	65

Introduction

This hands-on tutorial presents design patterns for collaborative data science using the featured precisionFDA pfda-ttyd, pfda-jupyterLab, and guacmole interactive workstation apps, and precisionFDA Databases. Through the development of the tutorial assets, precisionFDA's powerful capabilities for secure sharing and analysis of FISMA-Moderate authorized data are clearly demonstrated, and users will be empowered to develop their own collaborative regulatory data science use cases.

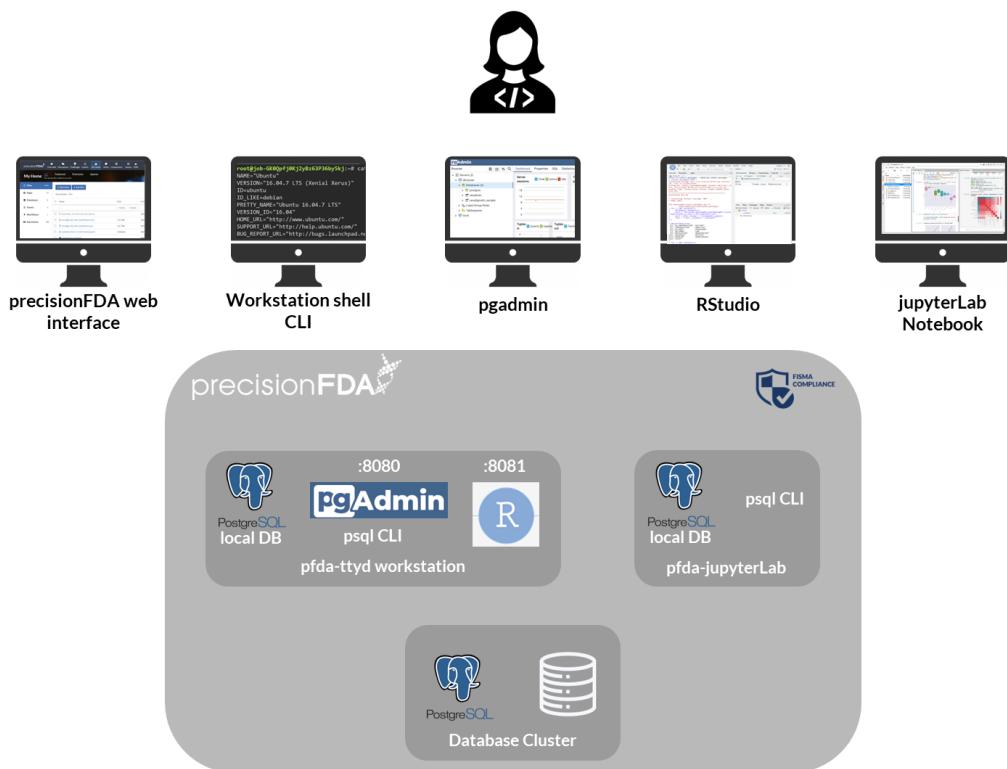
As always, keep in mind that all of these workstations, notebooks, and databases are strictly within the sole provenance of the user that launched them, and that in compliance with precisionFDA's FISMA authorization, the ability to deliver multi-user web services or databases is specifically not supported on precisionFDA.

Users can however use the power of the cloud to efficiently achieve their collaborative data science and bioinformatics objectives, and use the regulatory-grade platform to share the tools and results with full chain of provenance tracking. For cross-cutting analysis across FDA datasets, users will need to bring their data into the FDA's Intelligent Data Lifecycle Ecosystem (FiDLE).

Learning Objectives

Through this hands-on tutorial you will:

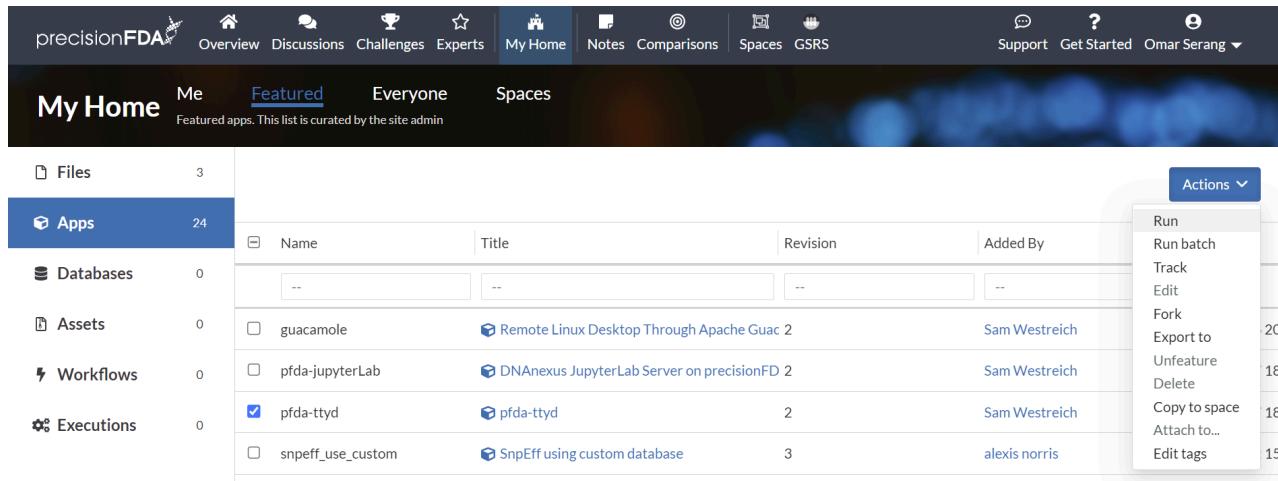
- Use the precisionFDA command line utility (pfda) to programmatically transfer files to and from precisionFDA and workstations and notebooks.
- Configure ttyd workstations to present multiple web services on ports 8080 using a reverse proxy for secure browser-based access with a rich UI.
- Launch a data analysis ttyd workstation with a local PostgreSQL database server, psql command line database client, pgadmin GUI database client, and RStudio configured with PostgreSQL access.
- Launch a SAS Studio workstation using a pfda-ttyd snapshot.
- Launch a KNIME Analytics Platform guacamole workstation with a local PostgreSQL database server, psql command line database client, and pgadmin GUI database client.
- Launch a precisionFDA Database cluster and access it from the data analysis workstation using psql and pgadmin to configure and install a database on the cluster from DDL and delimited data files.
- Use pgadmin to backup the cluster database to a precisionFDA file.
- Use pgadmin to restore the database backup to the data analysis workstation local database.
- Access the cluster and the workstation local databases from RStudio.
- Launch a jupyterLab workstation with a local PostgreSQL database server, and psql command line database client, and an example Python database analysis notebook.
- Launch a series of epidemiology-related Jupyter notebooks and use papermill to execute a long-running notebook non-interactively,
- Use pgadmin to restore the database backup to the jupyterLab workstation local database.
- Access the cluster and the workstation local databases from a Python notebook.



Build Data Analysis Workstation

Run the pfda-ttyd Featured App

Using the smallest instance type, run the Data Analysis Workstation job using the pfda-ttyd featured app.



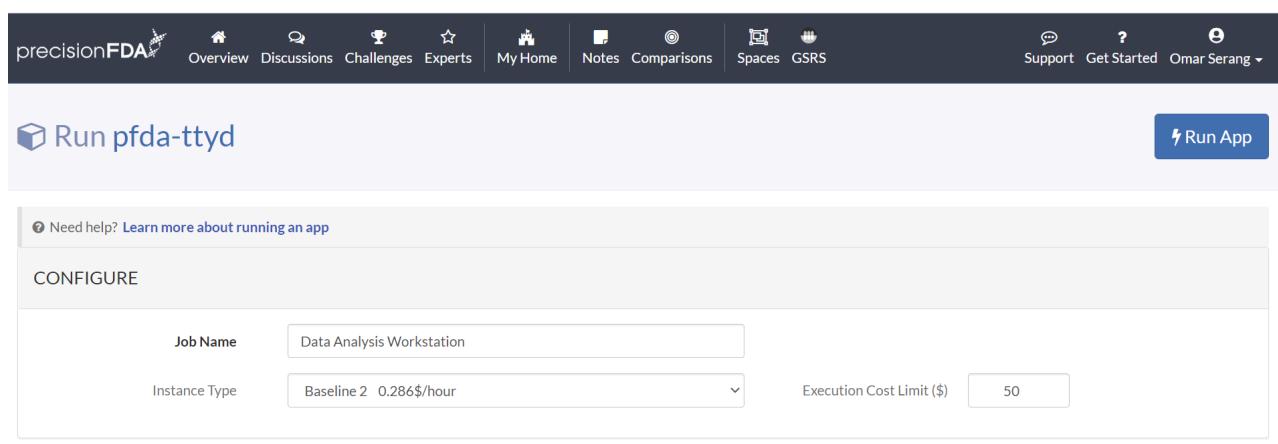
My Home Me Featured Everyone Spaces

Featured apps. This list is curated by the site admin

	Files	3	
Apps	24		
Databases	0		
Assets	0		
Workflows	0		
Executions	0		

Actions

- Run
- Run batch
- Track
- Edit
- Fork
- Export to
- Unfeature
- Delete
- Copy to space
- Attach to...
- Edit tags



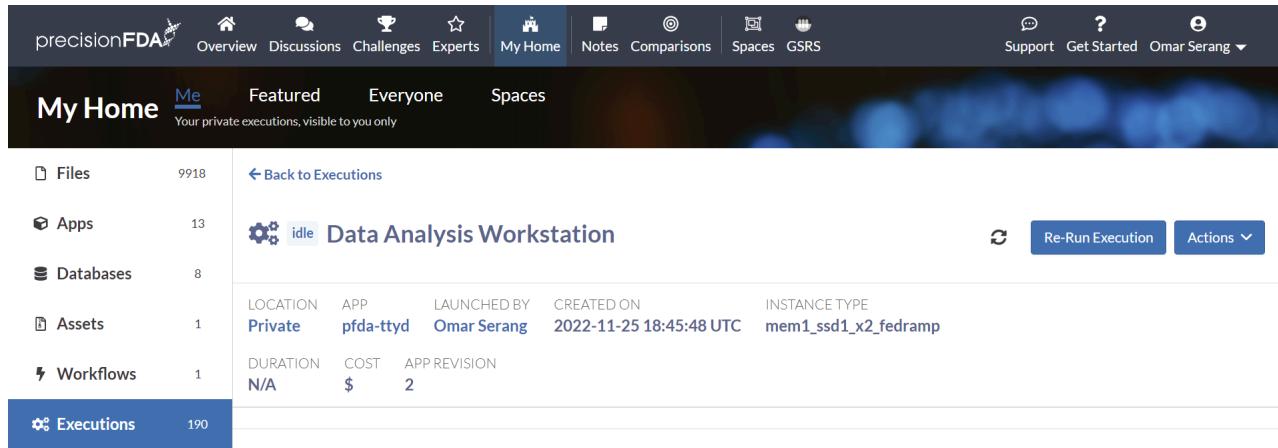
Run pfda-ttyd

Need help? [Learn more about running an app](#)

CONFIGURE

Job Name	Data Analysis Workstation
Instance Type	Baseline 2 0.286\$/hour
Execution Cost Limit (\$)	

50



My Home Me Featured Everyone Spaces

Your private executions, visible to you only

	Files	9918	
Apps	13		Back to Executions
Databases	8		
Assets	1		
Workflows	1		
Executions	190		

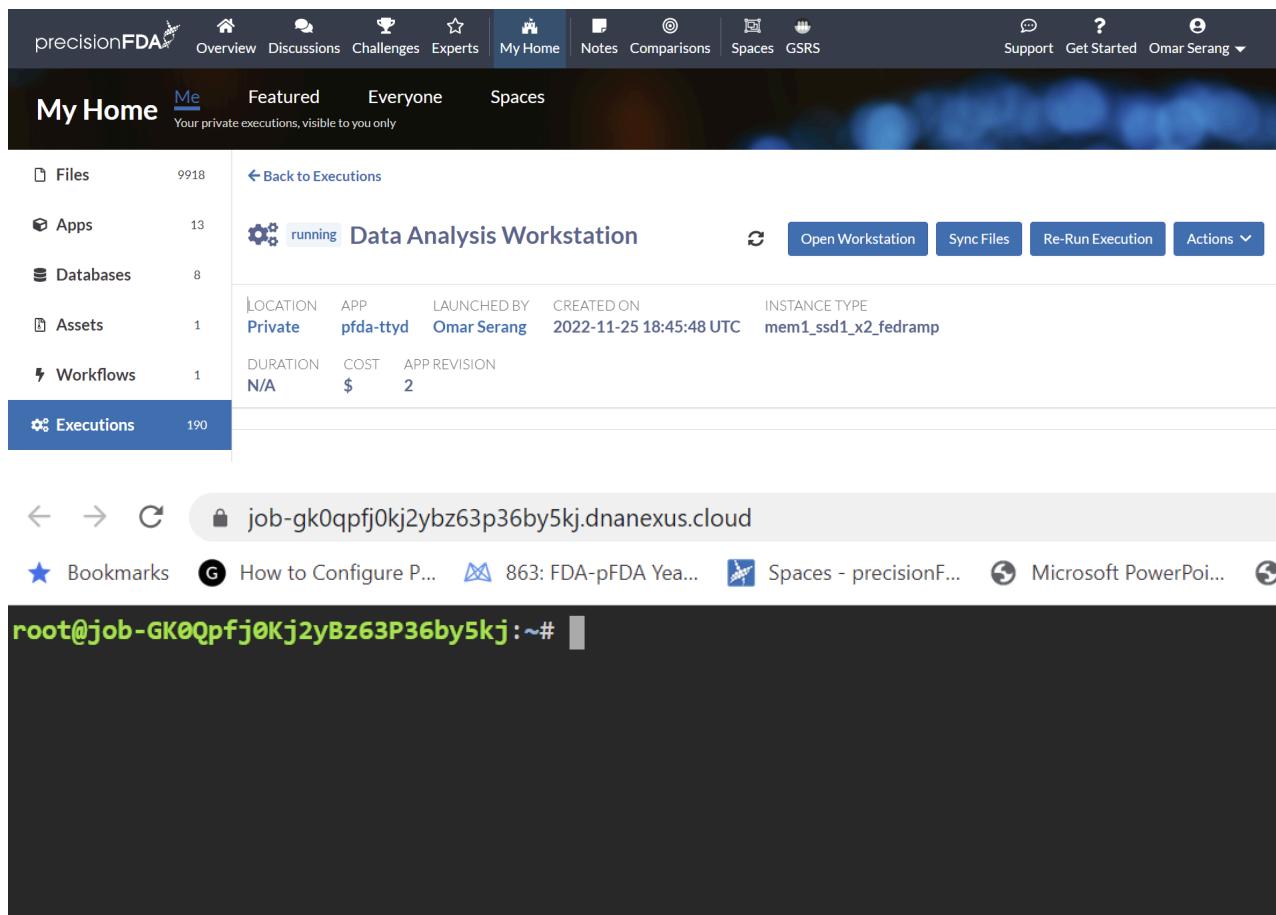
idle Data Analysis Workstation

LOCATION	APP	LAUNCHED BY	CREATED ON	INSTANCE TYPE
Private	pfda-ttyd	Omar Serang	2022-11-25 18:45:48 UTC	mem1_ssdl_x2_fedramp
DURATION	COST	APP REVISION		
N/A	\$	2		

Actions

- Re-Run Execution
- Actions

Refresh the execution status using the  button until the job is running and open the workstation.



The screenshot shows the precisionFDA interface. At the top, there's a navigation bar with links like Overview, Discussions, Challenges, Experts, My Home, Notes, Comparisons, Spaces, and GSRS. On the right, there are buttons for Support, Get Started, and a user profile for 'Omar Serang'. Below the navigation is a header bar with tabs for My Home, Me, Featured, Everyone, and Spaces. A message says 'Your private executions, visible to you only'. The main content area has a sidebar on the left with categories: Files (9918), Apps (13), Databases (8), Assets (1), Workflows (1), and Executions (190). The 'Executions' tab is selected. It displays a single execution entry for a 'Data Analysis Workstation' with details: LOCATION Private, APP pfda-ttyd, LAUNCHED BY Omar Serang, CREATED ON 2022-11-25 18:45:48 UTC, INSTANCE TYPE mem1_ssdl_x2_fedramp. Below this, it shows DURATION N/A, COST \$, and APP REVISION 2. To the right of the execution details are buttons for Open Workstation, Sync Files, Re-Run Execution, and Actions. At the bottom, there's a terminal window showing a root prompt on a server: 'root@job-GK0Qpfj0Kj2yBz63P36by5kj:~#'. Above the terminal, there's a search bar with the URL 'job-gk0qpfj0kj2ybz63p36by5kj.dnanexus.cloud' and a refresh button.

Use `dx-set-timeout` and `dx-get-timeout` to view and set the workstation application time-to-live after which it will self-terminate.

```
dx-set-timeout 2d
dx-get-timeout
0 days 23 hours 59 minutes 56 seconds
```

Present a simple web server on port 8080

The ttyd and guacamole workstations enable presentation of web services that can be accessed via the job URL port 8080. Let's startup a simple Python-based web server and browse to it.

```
python3 -m http.server 8080 &
```

Now copy the URL from your ttyd window and append port 8080 to it(e.g. <https://job-gxjjzj80kj2qgp2z6p9x1yvq.dnanexus.cloud:8080>) to browse to your web service on the workstation.



A screenshot of a web browser window. The address bar shows the URL: job-gk0qpfj0kj2ybz63p36by5kj.dnanexus.cloud:8080. Below the address bar, there are several tabs: Bookmarks, How to Configure P..., 863: FDA-pFDA Yea..., Spaces - precisionF..., and another tab that is partially visible. The main content area displays a directory listing for the root directory ('/'). The files listed are: .bash_history, .bash_logout, .bash_profile, .bashrc, .byobu/, .dnanexus_config/, .dx.timeout, .pfda_config, .profile, dnanexus-executable.json, dnanexus-job.json, and two dots (...).

Directory listing for /

- [.bash_history](#)
- [.bash_logout](#)
- [.bash_profile](#)
- [.bashrc](#)
- [.byobu/](#)
- [.dnanexus_config/](#)
- [.dx.timeout](#)
- [.pfda_config](#)
- [.profile](#)
- [dnanexus-executable.json](#)
- [dnanexus-job.json](#)
- [...](#)

Kill the http.server job to free up port 8080.

Deploy PostgreSQL, psql, pgadmin, and RStudio

A docker-compose.yml file is configured to launch a local PostgreSQL database, and RStudio, and pgadmin web services that connect to the database. RStudio and pgadmin are accessed using the workstation job URL extended to target the specific service on port 8080 (e.g.

<https://job-gb1y8jj0kj2xvbggbp3qgv55.dnanexus.cloud:8080/pgadmin/>). The postgres-local database container is accessed from the pgadmin and rstudio containers and from the psql command line client on the workstation shell.

The pfda-ttyd and guacamole workstation apps provide port 8080 for access to user deployed web services on the workstation. In order to access multiple different web services (e.g. pgadmin and RStudio) on the single port 8080, a Docker-based reverse proxy is configured using the traefik open source cloud-native application proxy application (<https://github.com/traefik/traefik>).

Install docker-compose-plugin.

```
install-docker-compose.sh
```

Download the traefik-postgres-pgadmin-rstudio-docker-compose.yml file to the workstation.

```
pfda download --file-id file-Gb21QGj0Kj2pBzz9Q3Yqv1Kp-2
```

Start the traefik, postgres-local, rstudio, and pgadmin docker services.

```
docker compose -f traefik-postgres-pgadmin-rstudio-docker-compose.yml
create
docker compose -f traefik-postgres-pgadmin-rstudio-docker-compose.yml
start
```

Install the psql postgres command line client and test the connection to the database.

```
apt update
apt install postgresql-client -y < "/dev/null"
```

```
PGPASSWORD="password" psql -h localhost -U postgres
```

Ctrl-D to exit psql.

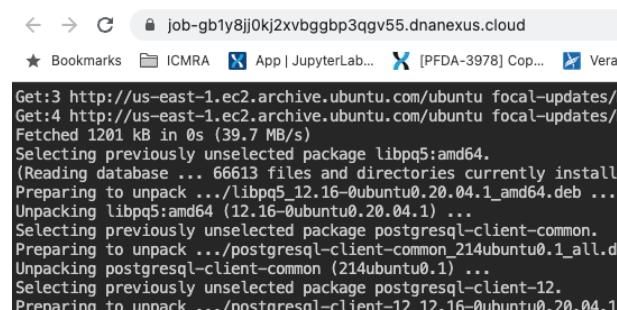
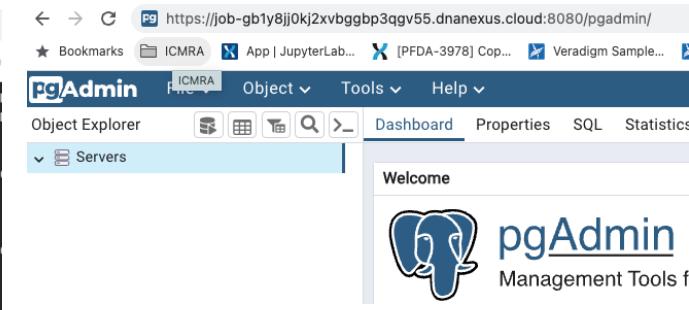
Configure the pgadmin container mounted directory permissions.

```
sudo chown -R 5050:5050 db_backups/
sudo chmod ugo+w db_backups/
```

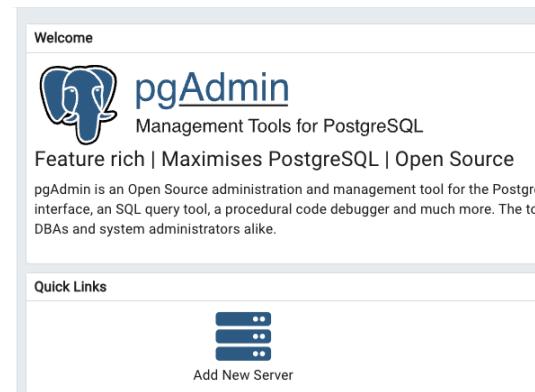
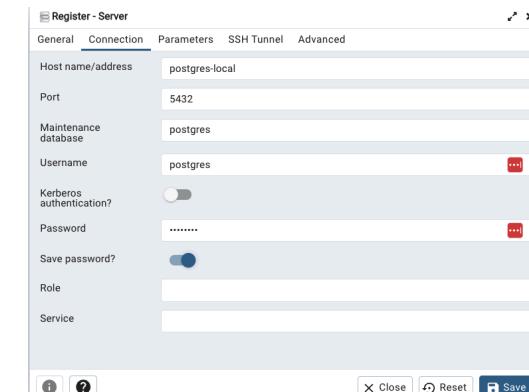
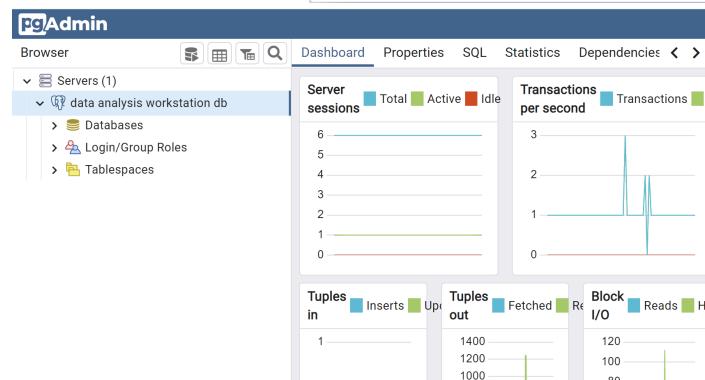
Access the postgres-local DB from pgadmin

Access the pgadmin web service from your web browser (e.g.

<https://job-gk0qpfj0kj2ybz63p36by5kj.dnanexus.cloud:8080/pgadmin>).

Add the workstation local database as a new server (e.g. Data Analysis Workstation DB Server) using hostname *postgres-local* and port *5432*, maintenance database *postgres*, user *postgres*, and password *password*.

Create a new database and tables

Connect to the cluster database from psql in the data analysis workstation shell.

```
PGPASSWORD="password" psql -h localhost -U postgres
```

Using psql, create a new database.

```
-- Database: workstations_and_databases_tutorial_db
CREATE DATABASE workstations_and_databases_tutorial_db
    WITH
        OWNER = postgres
        ENCODING = 'UTF8'
        CONNECTION LIMIT = -1
        IS_TEMPLATE = False;
```

Connect to the new database and create two tables.

```
\c workstations_and_databases_tutorial_db;

psql (9.5.25, server 11.16)
WARNING: psql major version 9.5, server major version 11.
          Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher:
ECDHE-RSA-AES128-GCM-SHA256, bits: 128, compression: off)
You are now connected to database
"workstations_and_databases_tutorial_db" as user "root".
workstations_and_databases_tutorial_db=>

CREATE TABLE public."PATIENT" (
    patient_id bigint NOT NULL,
    name character varying,
    gender character varying,
    zip character varying,
    country character varying,
    created_date date
);

CREATE TABLE public."OBSERVATION" (
    observation_id bigint NOT NULL,
    patient_id bigint,
    observation_name character varying,
    loinc character varying,
    created_date date
);

\dt
      List of relations
 Schema |     Name      | Type  | Owner
-----+-----+-----+-----
  public | OBSERVATION | table | root
  public | PATIENT    | table | root
(2 rows)
```

Load the cluster database from delimited text files

In the data analysis workstation shell, create a datafiles directory

```
mkdir datafiles
cd datafiles
```

Create file `patients.txt` with the following content:

```
cat > patients.txt
12345|Fred Foobar|M|94040|USA|2022-10-25
12346|Mary Merry|F|94040|USA|2022-09-24
12347|Barney Rubble|M|94040|USA|2022-08-23
ctrl-D
```

Create file `observations.txt` with the following content:

```
cat > observations.txt
9870|12345|Annual check up|66678-4|2022-11-01
9871|12345|Emergency|LG32756-5|2022-11-02
9872|12346|Clinic visit|66678-4|2022-11-03
9873|12347|Lab results|74418-5|2022-11-04
9874|12347|Post-op checkup|65375-8|2022-11-05
ctrl-D
```

[Copy the data into the cluster DB tables](#)

```
PGPASSWORD="password" psql -h localhost -U postgres
workstations_and_databases_tutorial_db=>
```

In psql:

```
\copy public."PATIENT" from '/home/dnanexus/datafiles/patients.txt'
delimiter '||' NULL ''

\copy public."OBSERVATION" from
'/home/dnanexus/datafiles/observations.txt' delimiter '||' NULL ''

select * from public."PATIENT";
patient_id | name | gender | zip | country | created_date
-----+-----+-----+-----+-----+-----+
-
12345 | Fred Foobar | M | 94040 | USA | 2022-10-25
12346 | Mary Merry | F | 94040 | USA | 2022-09-24
12347 | Barney Rubble | M | 94040 | USA | 2022-08-23

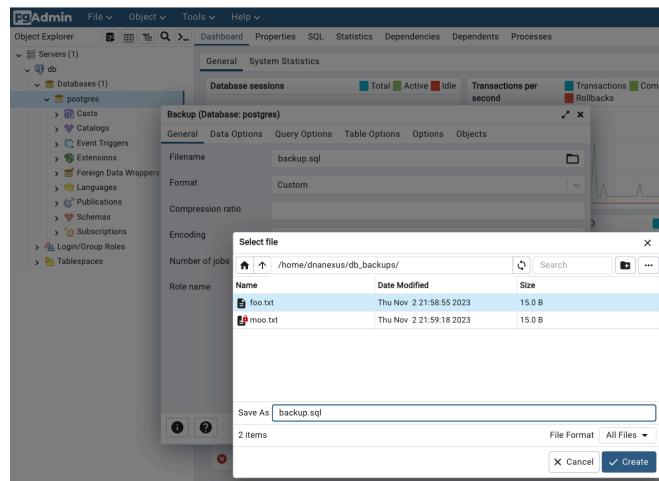
select * from public."OBSERVATION";
observation_id | patient_id | observation_name | loinc
|created_date
-----+-----+-----+-----+-----+
-
9870 | 12345 | Annual check up | 66678-4 |
2022-11-01
9871 | 12345 | Emergency | LG32756-5 |
2022-11-02
9872 | 12346 | Clinic visit | 66678-4 |
2022-11-03
9873 | 12347 | Lab results | 74418-5 |
2022-11-04
9874 | 12347 | Post-op checkup | 65375-8 |
2022-11-05
```

Observe the new tables and data in the pgadmin Workstations and Databases Tutorial server connection.

Share Files Between Workstation FS and PostgreSQL Docker FS

Since pgadmin is running in a Docker container on the workstation, we are going to have to connect to the pgadmin container shell and copy files we want to share between the workstation and pgadmin to and from the mount point shared by the container and the workstation (i.e. /home/dnanexus/db_backups).

When performing a database backup in pgadmin, save the file in /home/dnanexus/db_backups.



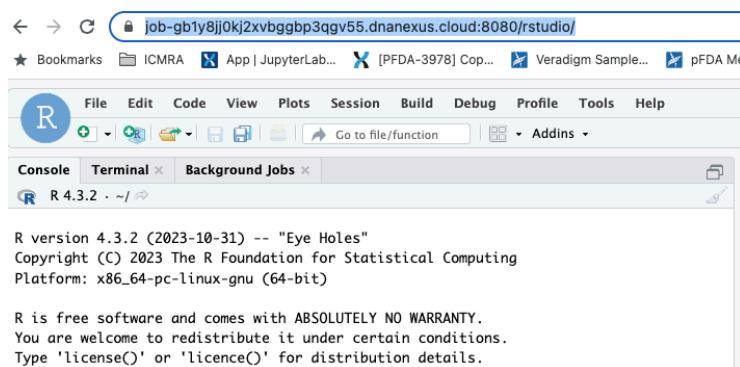
The backup file is accessible in workstation /home/dnanexus/db_backups. Files can be placed in this directory for access by pgadmin.

```
root@job-Gb21Kfj0Kj2jxbjv0pzxQx8Y:~# ls /home/dnanexus/db_backups/
backup.sql  foo.txt  moo.txt
```

Access the postgres-local DB from rstudio

Access the RStudio web service from your web browser (e.g.

<https://job-gb1y8jj0kj2xvbgbp3qgv55.dnanexus.cloud:8080/rstudio/>.



```
R version 4.3.2 (2023-10-31) -- "Eye Holes"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

Install the RPostgres packages and test access to the workstation local database. In the R Studio console:

```
# Install RPostgres
```

```
install.packages("RPostgres")

# Connect to local database
library(DBI)
con <- DBI::dbConnect(
  RPostgres::Postgres(),
  host = "172.17.0.1",
  port = 5432, dbname = "postgres",
  user = "postgres", password = "password"
)
# List the tables in db postgres
dbListTables(con)
```

Since there are no tables in the postgres database, the response is `character(0)`. Let's add two tables using psql and run the same R query.

Using the psql command line client in the data analysis workstation shell:

```
PGPASSWORD="password" psql -h localhost -U postgres

CREATE TABLE public."PATIENT" (
    patient_id bigint NOT NULL,
    name character varying,
    gender character varying,
    zip character varying,
    country character varying,
    created_date date
);

CREATE TABLE public."OBSERVATION" (
    observation_id bigint NOT NULL,
    patient_id bigint,
    observation_name character varying,
    loinc character varying,
    created_date date
);
```

The same query in the RStudio console now shows the two new tables.

```
dbListTables(con)
[1] "PATIENT"      "OBSERVATION"
```

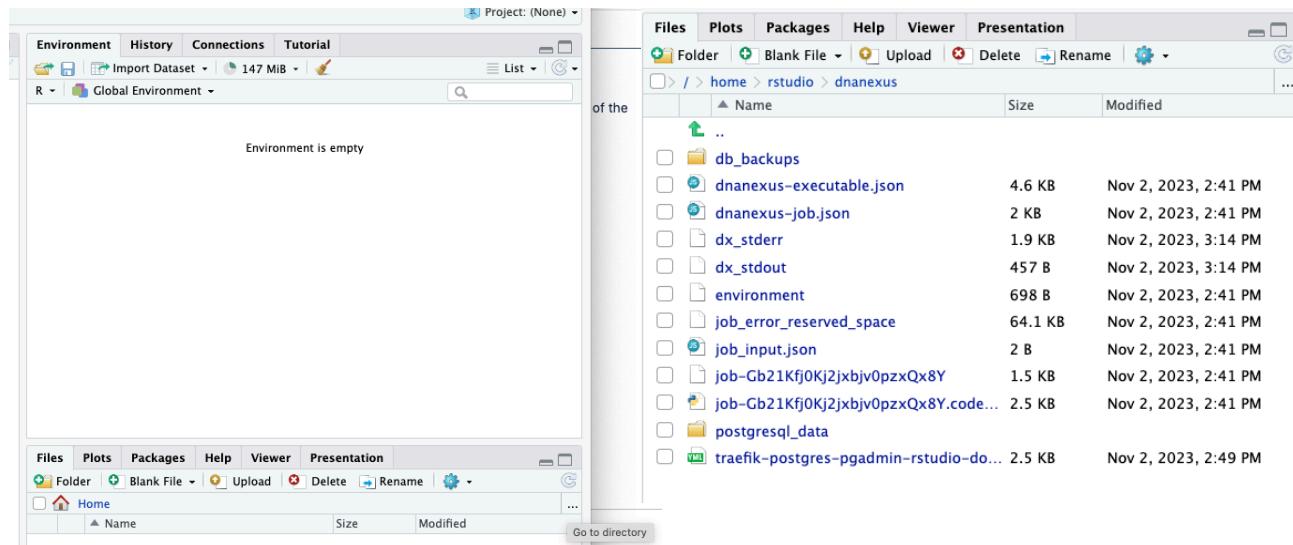
Let's drop the tables since we will be populating this database from a backup a later stage in this tutorial. Using the psql command line client in the data analysis workstation shell:

```
DROP TABLE public."PATIENT"
DROP TABLE public."OBSERVATION"
```

Control-D to exit psql.

Share Files Between Workstation FS and RStudio Docker FS

The rstudio container shares the /home/dnanexus mount point with the workstation filesystem so it is straightforward to access the workstation filesystem from within RStudio. Simply set the path in the RStudio file browser to /home/rstudio/dnanexus.



PostgreSQL Tips

Force drop connections

If you need to drop a database, you'll need to close all the sessions connected to it using the following PostgreSQL code:

```

SELECT
    pg_terminate_backend(pid)
FROM
    pg_stat_activity
WHERE
    -- don't kill my own connection!
    pid <> pg_backend_pid()
    -- don't kill the connections to other databases
    AND datname = 'ehr_data'
;
    
```

Quick queries for estimated row counts

To present the estimated row count for each table:

```

SELECT relname, TO_CHAR(n_live_tup, 'fmG999G999G999')
    FROM pg_stat_user_tables
ORDER BY relname ASC;
    
```

Alternatively:

```

SELECT
    pgClass.relname AS tableName,
    TO_CHAR(pgClass.reltuples, 'fmG999G999G999') AS rowCount
FROM
    pg_class pgClass
    
```

```

INNER JOIN
    pg_namespace pgNamespace ON (pgNamespace.oid =
pgClass.relnamespace)
WHERE
    pgNamespace.nspname NOT IN ('pg_catalog', 'information_schema') AND
    pgClass.relkind='r'
ORDER BY pgClass.reltuples ASC

```

To present to total row count for all tables:

```

SELECT TO_CHAR(SUM(n_live_tup), 'fmG999G999G999')
FROM pg_stat_user_tables

```

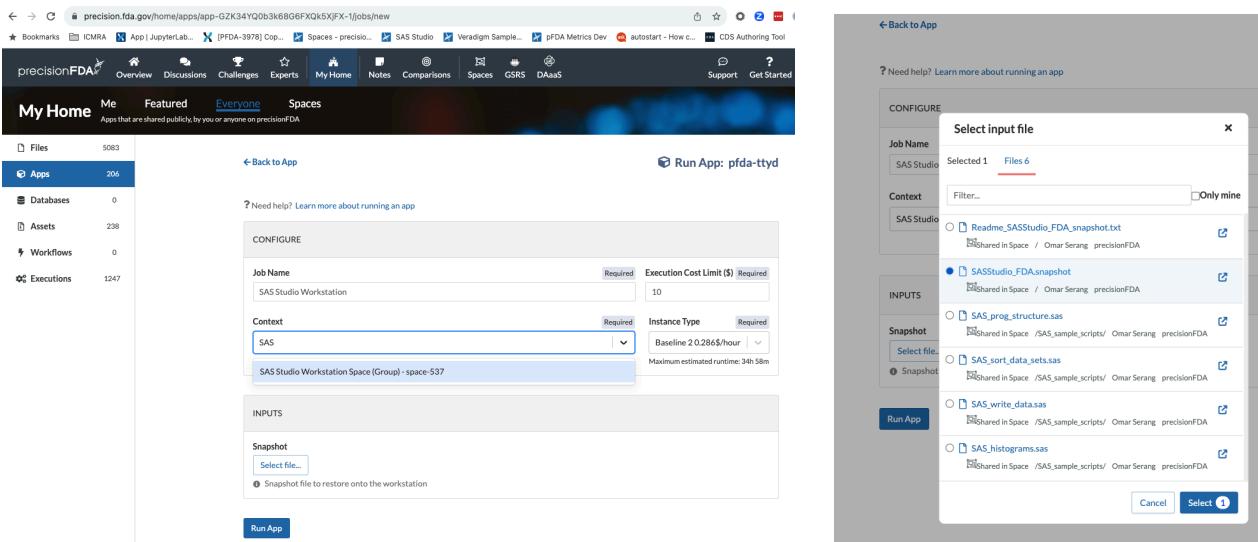
Build SAS Studio Workstation

SAS Studio Workstation Space

You will need to be a member of the SAS Studio Workstation Space (<https://precision.fda.gov/spaces/537>) in order to use the FDA's enterprise SAS license that is incorporated into a SAS Studio Snapshot file located in the Space (e.g. SASStudio_FDA.snapshot). FDA users that wish to use SAS Studio on precisionFDA should contact precisionFDA Support to request membership in the SAS Studio Workstation Space.

Run the pfda-ttyd Featured App with SAS Studio Snapshot

Follow the procedure in the [Run the pfda-ttyd Featured App](#) section of this tutorial, selecting the execution Context as SAS Studio Workstation Space and specifying SASStudio_FDA.snapshot in the Snapshot inputs section.



The screenshot shows the precisionFDA web interface. On the left, the 'My Home' dashboard displays various metrics: 5082 Files, 206 Apps, 0 Databases, 238 Assets, 0 Workflows, and 1247 Executions. In the center, a modal window titled 'Run App: pfda-ttyd' is open. It has two tabs: 'CONFIGURE' and 'INPUTS'. Under 'CONFIGURE', there are fields for 'Job Name' (set to 'SAS Studio Workstation'), 'Execution Cost Limit (\$)' (set to '10'), 'Context' (set to 'SAS'), 'Instance Type' (set to 'Baseline 2.0286\$/hour'), and 'Snapshot' (with a dropdown menu). Under 'INPUTS', there is a 'Snapshot' section with a 'Select file...' button. To the right of the modal, a detailed view of the 'Select input file' dialog is shown, listing several SAS files from the 'SAS Studio' context, including 'Readme_SASStudio_FDA_snapshot.txt', 'SASStudio_FDA.snapshot', 'SAS_prog_structure.sas', 'SAS_sort_data_sets.sas', 'SAS_write_data.sas', and 'SAS_histograms.sas'. The 'SASStudio_FDA.snapshot' file is selected.

Open the workstation once it is running, download the sample SAS code, and start SAS Studio.

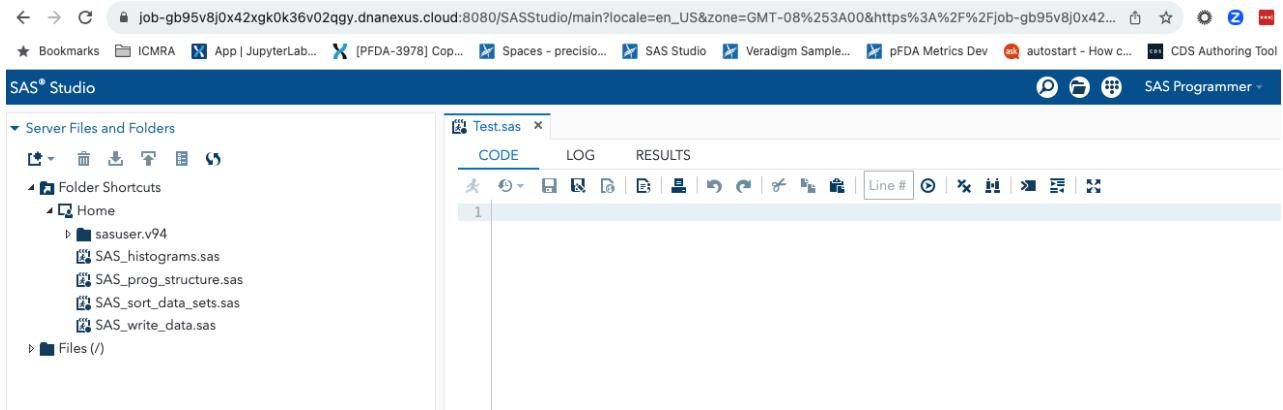
```

pfda download -space-id 537 -folder-id 8331897
mv *.sas /home/sasuser/
mv *.xpt /home/sasuser/
mkdir /home/sasuser/SAS_Datasets
chown sasuser /home/sasuser/SAS_Datasets/

```

```
./sasstudio.sh start
```

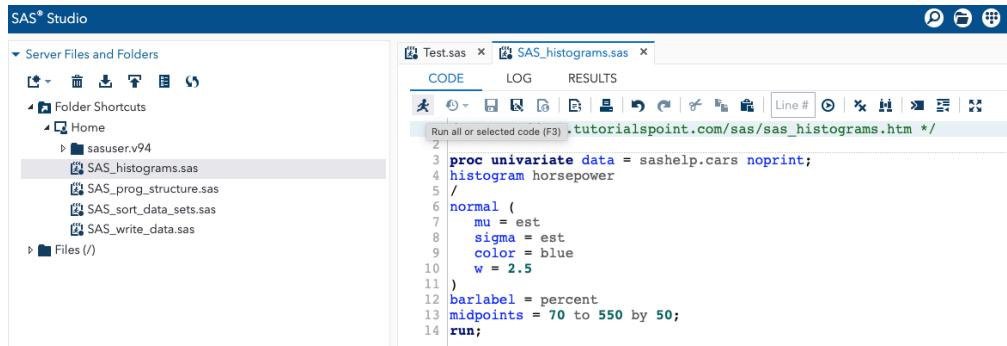
Copy the ttyd job URL and append :8080/SASStudio to it (e.g. <https://job-gb95v8j0x42xgk0k36v02qgy.dnanexus.cloud:8080/SASStudio>) to open SAS Studio and login with user "sasuser" password "sas".

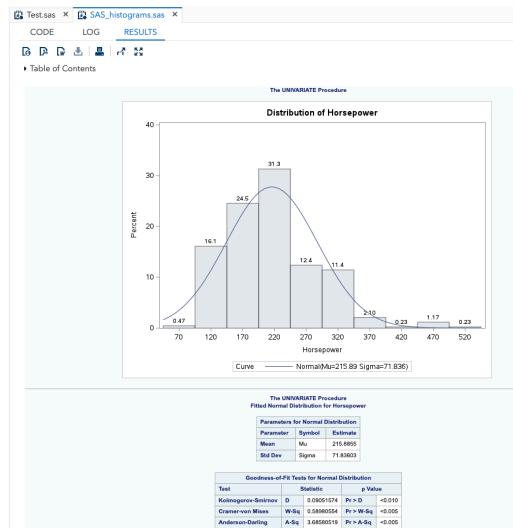


Note that the home folder in SAS Studio is mapped to /home/sasuser on the workstation local filesystem.

```
ls /home/sasuser/
SAS_histograms.sas  SAS_prog_structure.sas  SAS_sort_data_sets.sas
SAS_write_data.sas  sasuser.v94
```

Open up one of the SAS example files and run it.





Open an SDTM File in SAS Studio

Open the Open_SDTM.sas app and run it to read a SDTM file in xport format into SAS Studio then select View Column Labels to explore the sample clinical data.

Study Identifier	Domain Abbreviation	Unique Subject Identifier	Sequence Number	Group ID	Sponsor-Defined Identifier
1 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	1		4601
2 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	2		4602
3 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	3		4603
4 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	4		4604
5 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	5		4606
6 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	6		4608
7 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	7		4610
8 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	8		4611
9 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	9		4613
10 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	10		4614
11 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	11		4633
12 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	12		4634
13 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	13		4635
14 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	14		4642
15 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	15		4643
16 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	16		12476
17 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	17		59
18 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	18		4714
19 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	19		12516
20 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	20		12522
21 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	21		12536
22 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	22		12538
23 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	23		12539
24 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	24		12540
25 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	25		12541
26 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	26		12553
27 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	27		12555
28 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	28		12568
29 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	29		12573
30 CDMH2-STUDY-637517d660e1a52a4f46860 PR		1021000010019464	30		12574

Build KNIME Workstation

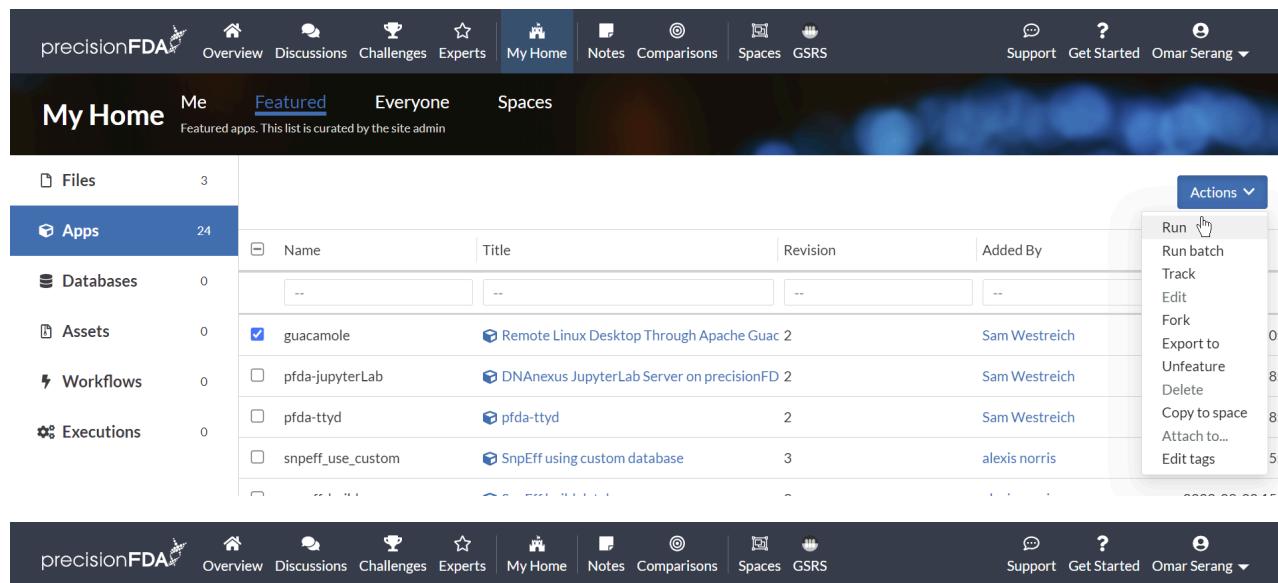
This tutorial demonstrates installation of the KNIME Analysis Platform as a desktop client application on a Guacamole workstation. The KNIME tutorial workflow:

- Creates a database on the local PostgreSQL server
- Downloads data files from a designated precisionFDA folder to the local filesystem
- ETLs the data from the local filesystem into the database
- Performs pivot analysis and geomap presentation of the data
- Uploads analysis results to precisionFDA My Home area

This provides KNIME examples for connecting to precisionFDA, running shell scripts, and executing DB operations using SQL.

Run the guacamole Featured App

Using the a baseline-4 instance type, run the KNIME Workstation job using the guacamole featured app. Specify a maximum session length of 5y.

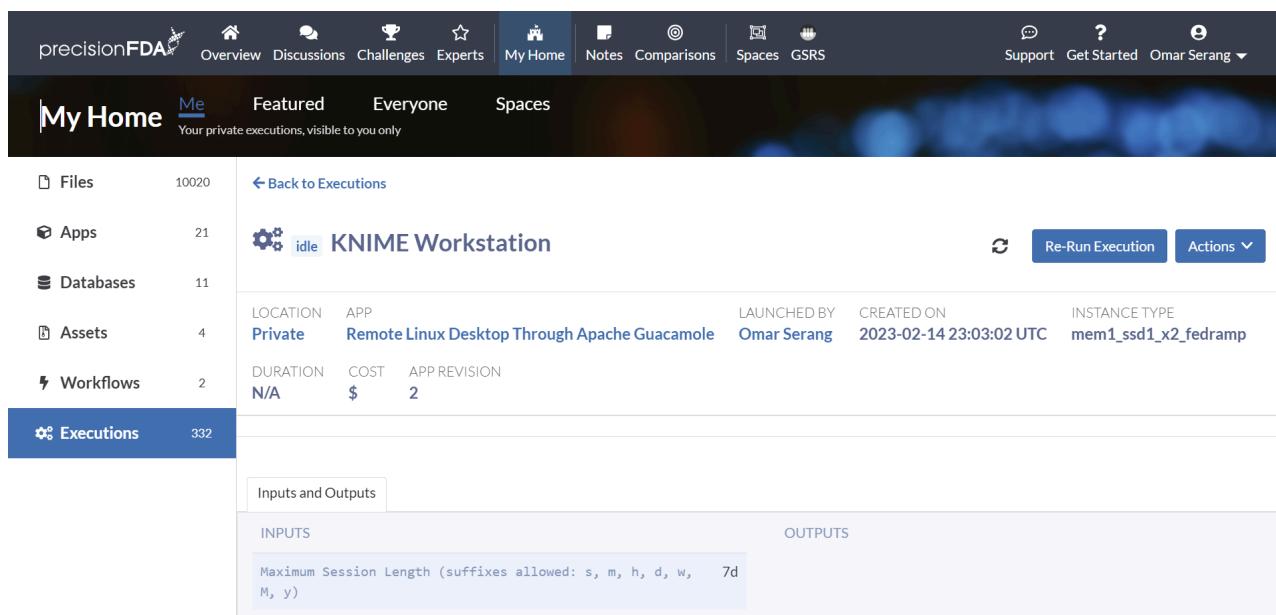


	Name	Title	Revision	Added By
<input checked="" type="checkbox"/>	guacamole	Remote Linux Desktop Through Apache Guac 2		Sam Westreich
<input type="checkbox"/>	pfda-jupyterLab	DNAnexus JupyterLab Server on precisionFD 2		Sam Westreich
<input type="checkbox"/>	pfda-ttyd	pfda-ttyd	2	Sam Westreich
<input type="checkbox"/>	snpeff_use_custom	SnpEff using custom database	3	alexis norris



? Need help? Learn more about running an app

CONFIGURE	
Job Name	Required
Execution Cost Limit (\$)	
Required	
100	
Instance Type	Required
Baseline 2 0.286\$/hour	

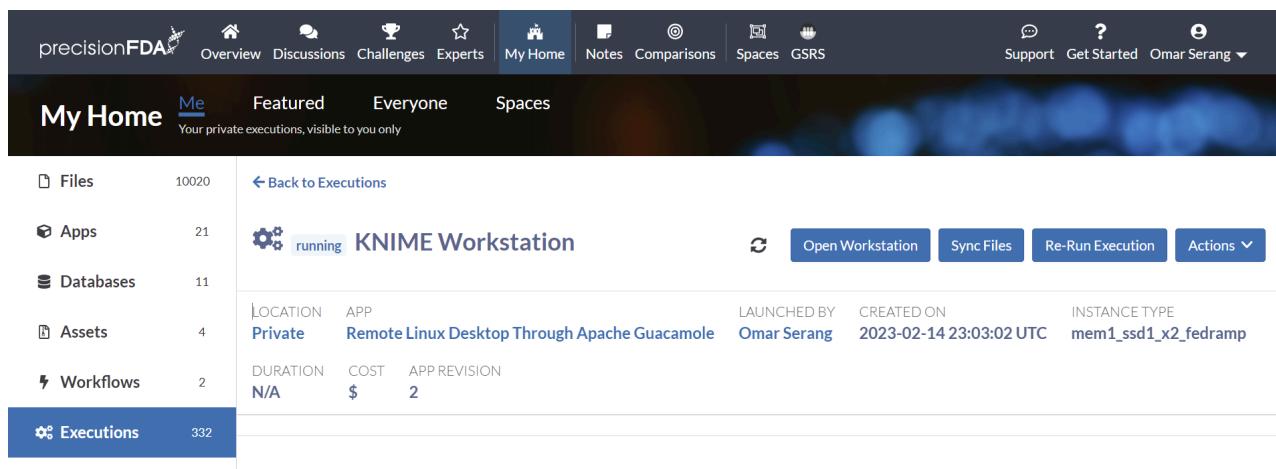


The screenshot shows the 'My Home' section of the precisionFDA platform. On the left sidebar, under 'Executions', there is a card for the 'KNIME Workstation'. The card displays the following information:

- LOCATION:** Private
- APP:** Remote Linux Desktop Through Apache Guacamole
- LAUNCHED BY:** Omar Serang
- CREATED ON:** 2023-02-14 23:03:02 UTC
- INSTANCE TYPE:** mem1_ssdl_x2_fedramp

Below this, there are sections for 'DURATION', 'COST', and 'APP REVISION' (N/A, \$, 2). At the bottom of the card, there is a 'Inputs and Outputs' section with tabs for 'INPUTS' and 'OUTPUTS'. A note states: 'Maximum Session Length (suffixes allowed: s, m, h, d, w, M, y) 7d'.

Refresh the execution status using the  button until the job is running and open the workstation. Note that it takes a few minutes for the guacamole workstation to come up after going into running status.

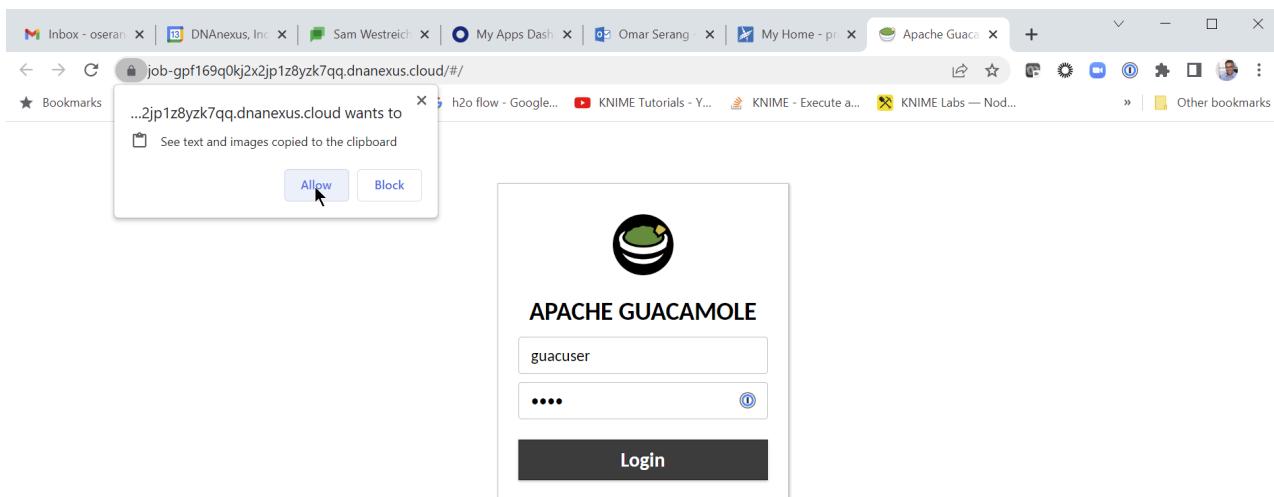


The screenshot shows the 'My Home' section of the precisionFDA platform. On the left sidebar, under 'Executions', there is a card for the 'KNIME Workstation'. The card displays the following information:

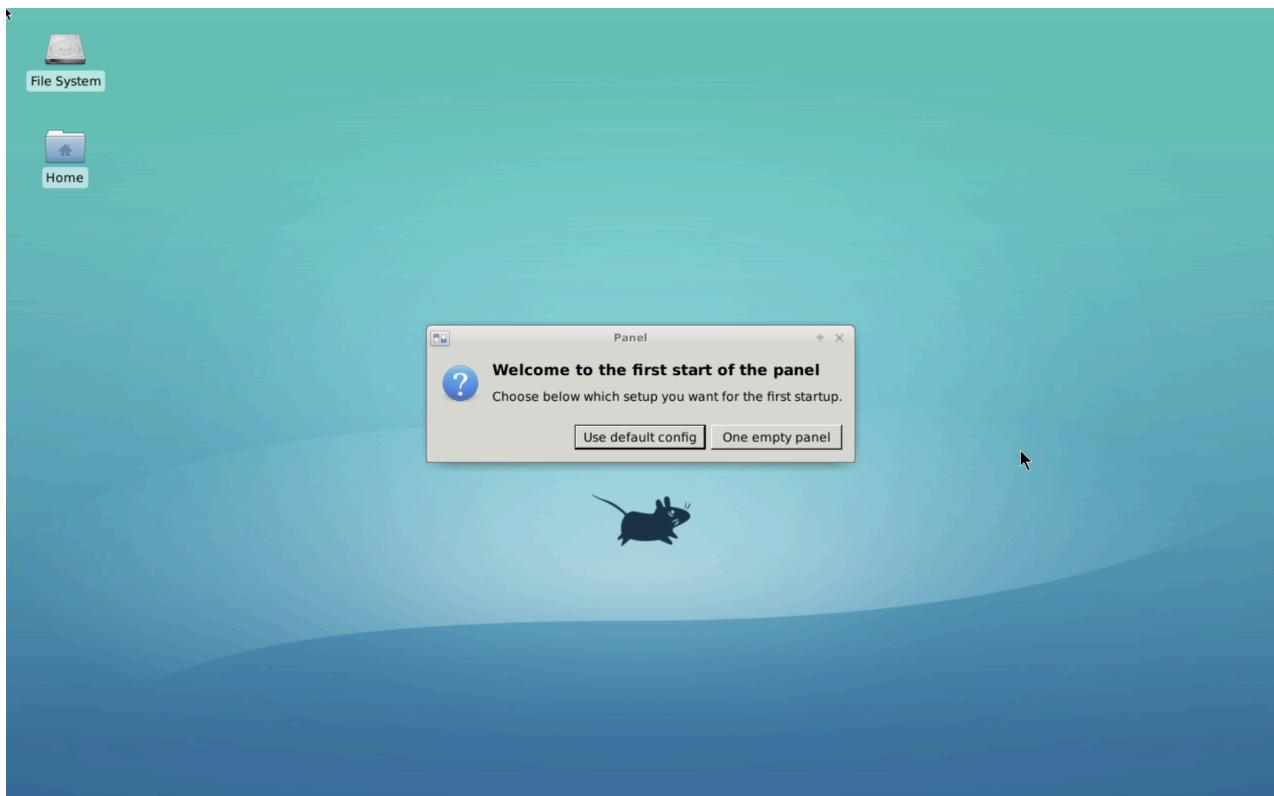
- LOCATION:** Private
- APP:** Remote Linux Desktop Through Apache Guacamole
- LAUNCHED BY:** Omar Serang
- CREATED ON:** 2023-02-14 23:03:02 UTC
- INSTANCE TYPE:** mem1_ssdl_x2_fedramp

Below this, there are sections for 'DURATION', 'COST', and 'APP REVISION' (N/A, \$, 2). At the bottom of the card, there is a 'Inputs and Outputs' section with tabs for 'INPUTS' and 'OUTPUTS'. A note states: 'Maximum Session Length (suffixes allowed: s, m, h, d, w, M, y) 7d'.

Allow the desktop to see text and images copied to the clipboard and login with user "guacuser" password "test".

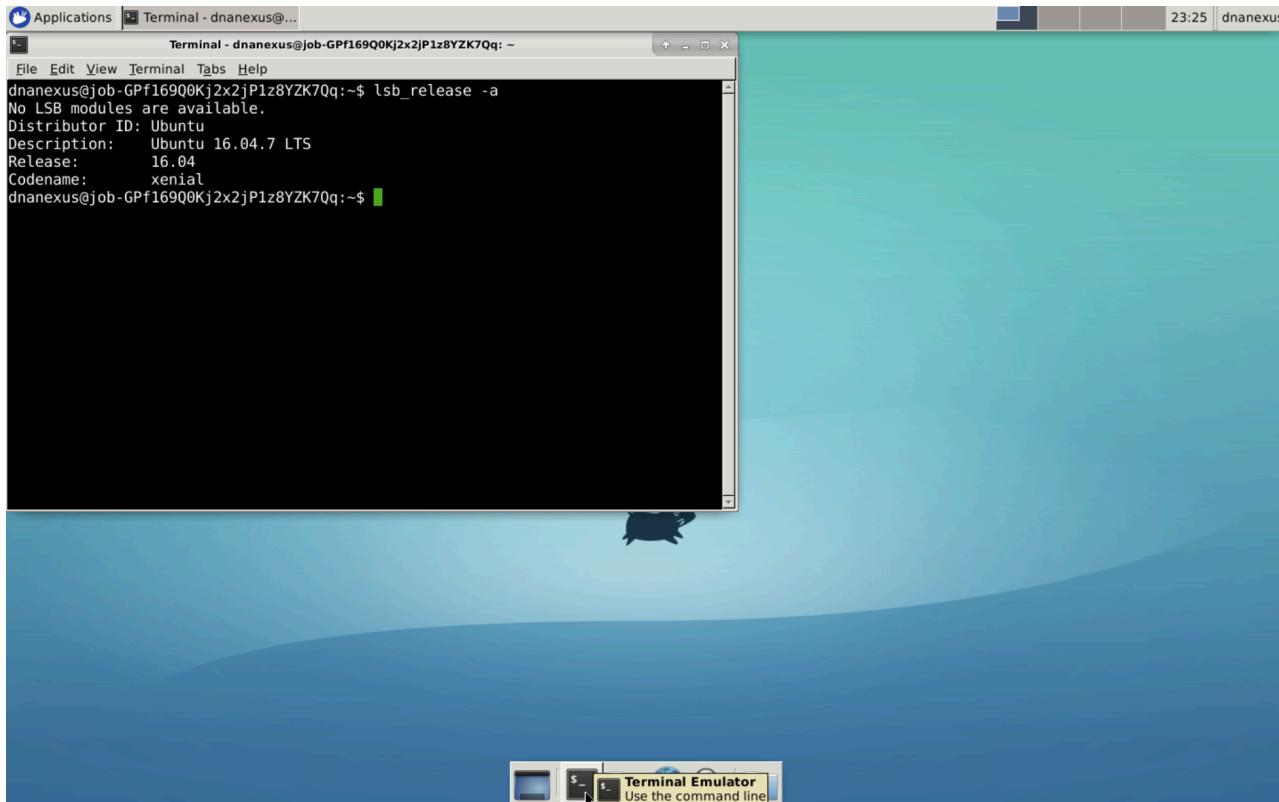


Use the default panel configuration when first entering the Linux desktop environment.



Open a terminal emulator window, check the OS version. Note that I needed to use ctrl-shift-v to paste from my laptop to the workstation.

```
lsb_release -a
```



Adjust environment variables to enable interaction with file on precisionFDA.

```
unset DX_WORKSPACE_ID
dx cd $DX_PROJECT_CONTEXT_ID
```

Use dx-get-timeout and dx-set-timeout to view and set the workstation application time-to-live after which it will self-terminate.

```
dx-set-timeout 5y
dx-get-timeout
```

Install Additional Utilities and Dependencies

```
# Browser, tree, dos2unix
sudo apt update
sudo apt-get install -y chromium-browser < "/dev/null"
sudo apt install -y tree < "/dev/null"
sudo apt install -y dos2unix < "/dev/null"

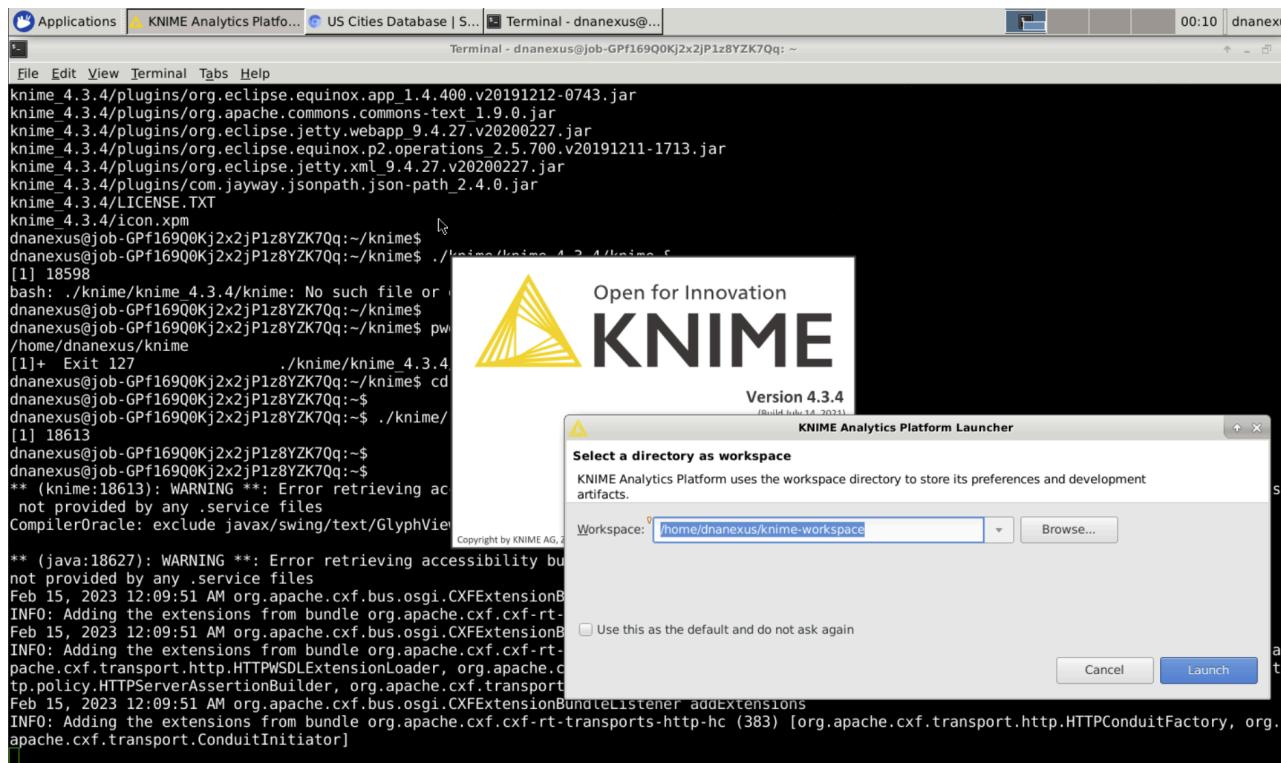
# KNIME Dependencies
sudo apt install -y libwebkit2gtk-4.0-37 < "/dev/null"
sudo apt install -y libgtk-3-dev < "/dev/null"
```

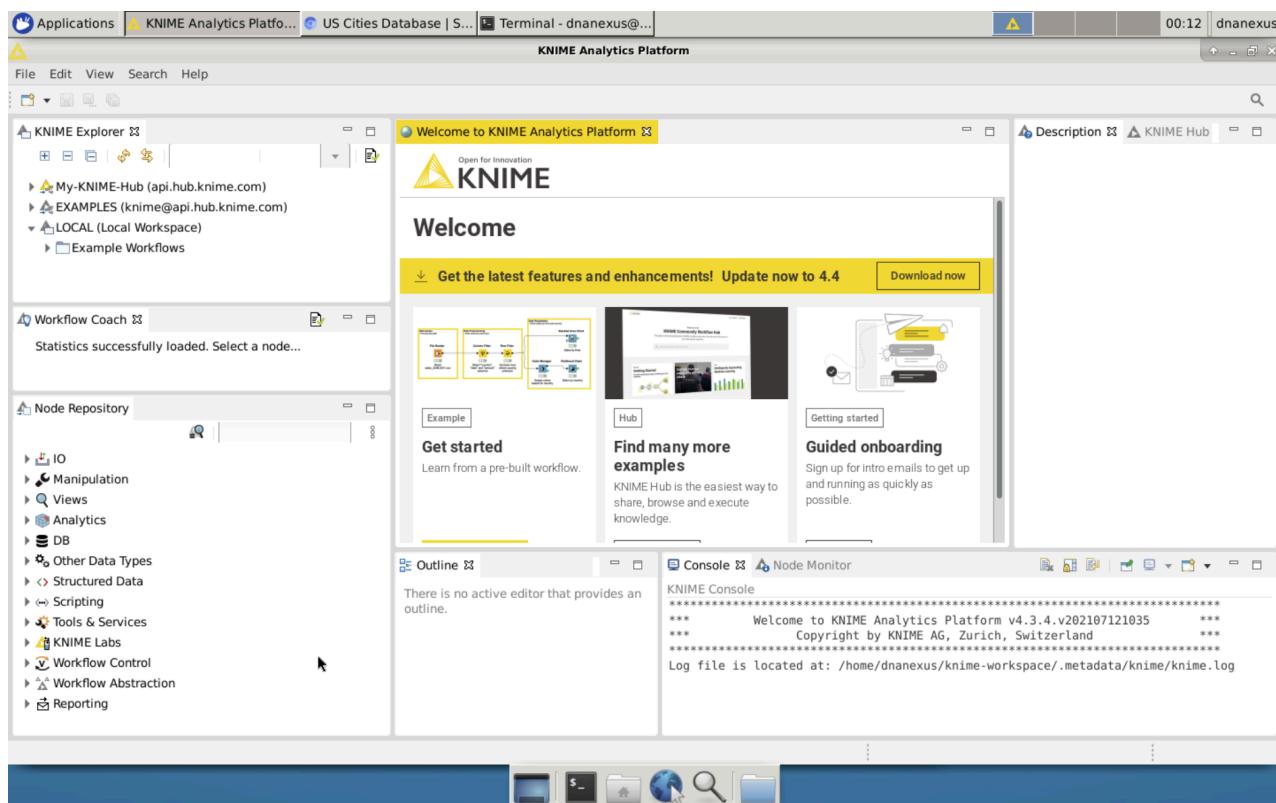
Install and Start KNIME

Install start KNIME and accept the default Workspace directory. Accept the offer to help improve KNIME since that will enable some of KNIME's wizard capabilities.

```
# KNIME
cd ~
```

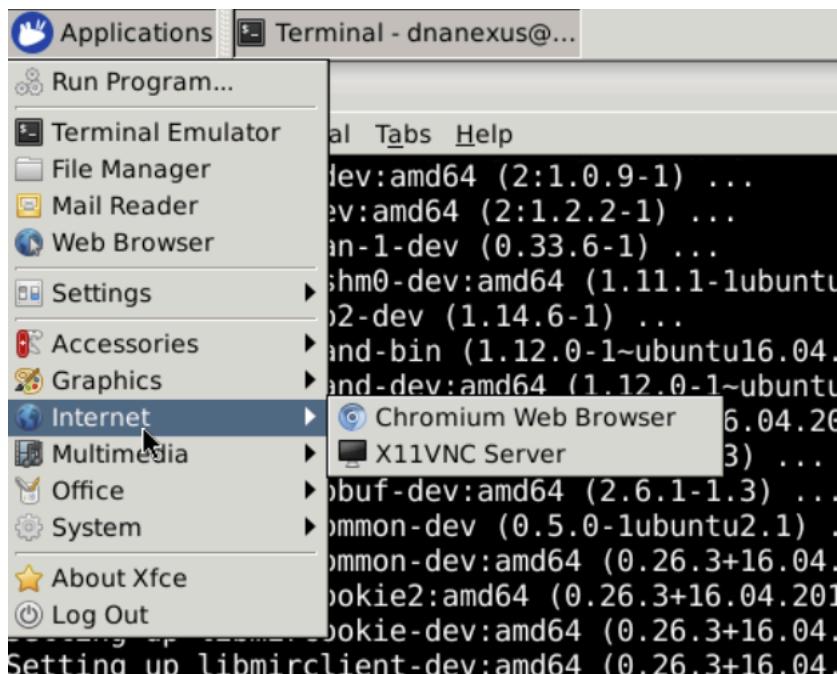
```
mkdir -p knime
cd knime
wget
https://download.knime.org/analytics-platform/linux/knime-latest-linu
x.gtk.x86_64.tar.gz
tar xvf knime-latest-linux.gtk.x86_64.tar.gz
cd
./knime/knime_4.7.3/knime &
```

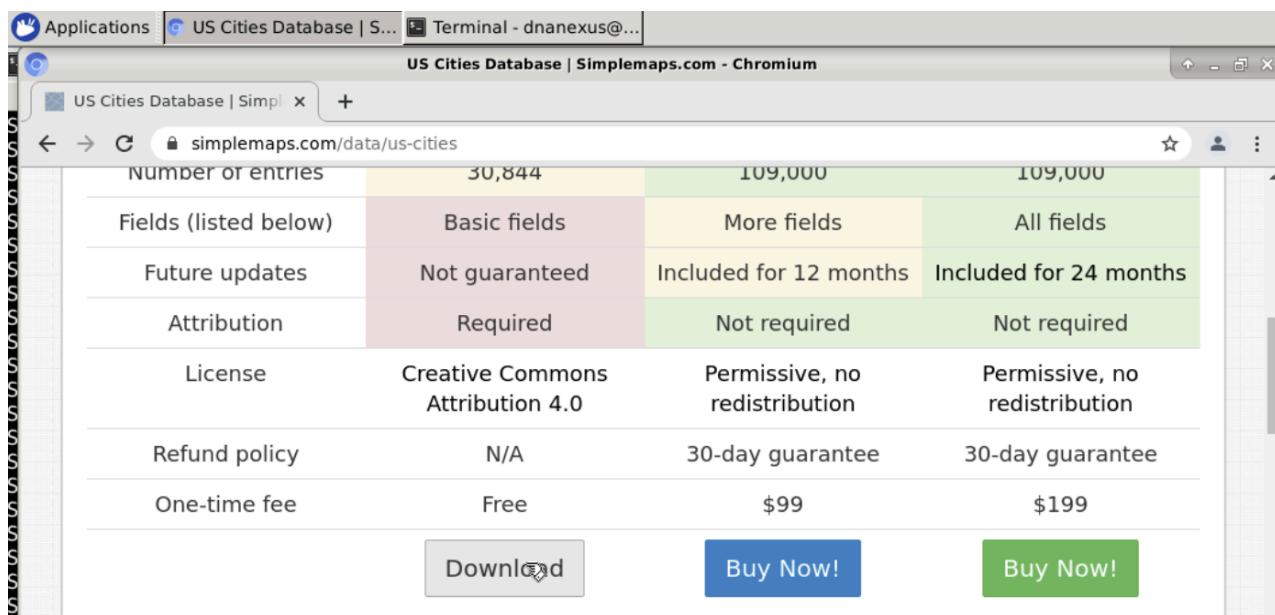




Install US City Geo Data Using the Chromium Browser

Start the Chromium Browser and download *simplemaps_uscities_basicv1.75.zip* from <https://simplemaps.com/data/us-cities>.





Number of entries	30,844	109,000	109,000
Fields (listed below)	Basic fields	More fields	All fields
Future updates	Not guaranteed	Included for 12 months	Included for 24 months
Attribution	Required	Not required	Not required
License	Creative Commons Attribution 4.0	Permissive, no redistribution	Permissive, no redistribution
Refund policy	N/A	30-day guarantee	30-day guarantee
One-time fee	Free	\$99	\$199

Leaving the previous terminal for KNIME to run in the background, start a new terminal window and set the key variable to the cli authentication token.

```
key=<copied key>

cd
mv Downloads/simplemaps_uscities_basicv1.76.zip .
unzip simplemaps_uscities_basicv1.76.zip
rm license.txt uscities.xlsx
mv uscities.csv knime-workspace/
rm simplemaps_uscities_basicv1.76.zip
```

Deploy Local PostgreSQL DB Server and CLI

Deploy a local PostgreSQL DB server on the Data Analysis workstation. Map the postgres port from the container to the workstation (host) OS. Note that there is already a dockerized PostgreSQL DB used by Guacamole so this will be a second instance.

```
# Install and start a second PostgreSQL server (and psql CLI)
# Note there is already a postgres docker container that is used by
guacamole
sudo docker run --name postgres2 -e POSTGRES_PASSWORD=password -p
5432:5432 -d postgres:13.4-buster

# Install postgres client
sudo apt update
sudo apt install -y postgresql-client < "/dev/null"

# Connect to local postgres db
PGPASSWORD="password" psql -h localhost -U postgres -c '\l'
```

Deploy pgadmin and Connect to the Local DB

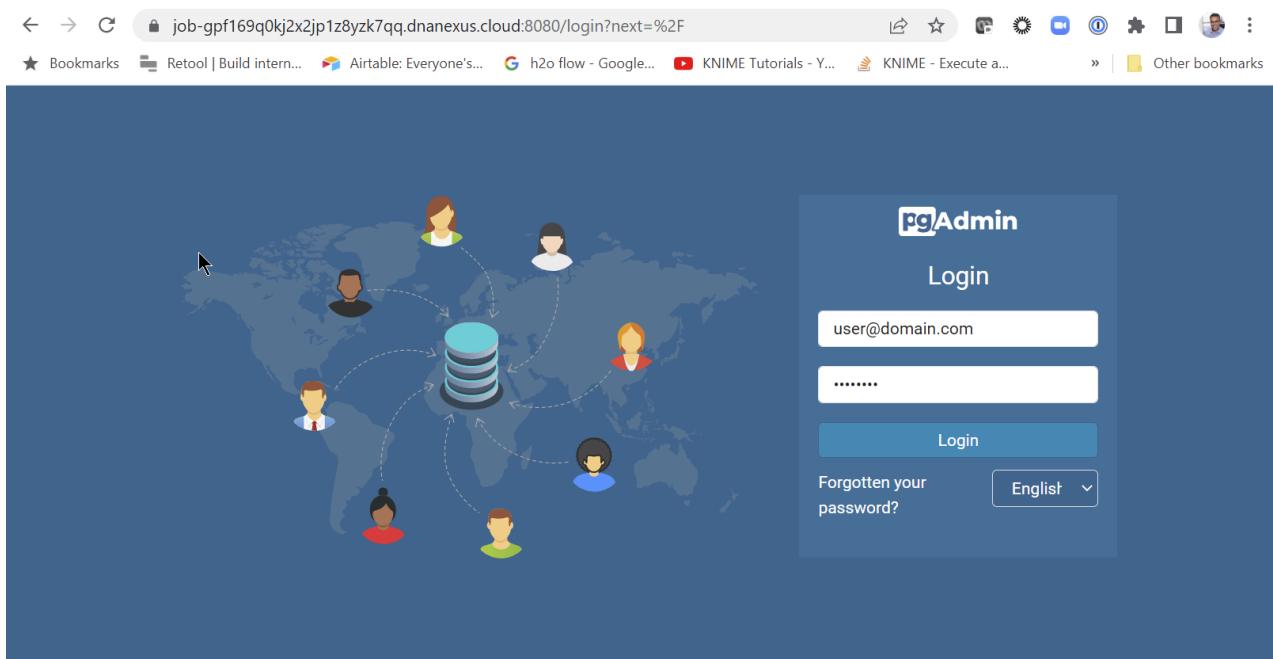
pgadmin4 is deployed in a Docker container mapping the pgadmin web service port 80 to workstation port 8080. A directory is created on the workstation with the appropriate ownership to enable database backup files created in pgadmin to be copied from the container to the workstation.

```
# Create and configure host directory for backup files from pgadmin
cd
mkdir /home/dnanexus/db_backups
sudo chown -R 5050:5050 db_backups/
sudo chmod ugo+w db_backups/

# Run pgadmin
sudo docker run --name pgadmin -it -v
/home/dnanexus/db_backups:/home/dnanexus/db_backups -p 8080:80 -e
'PGADMIN_DEFAULT_EMAIL=user@domain.com' -e
'PGADMIN_DEFAULT_PASSWORD=password' -d dpage/pgadmin4
```

Access the pgadmin web service from your web browser (e.g.

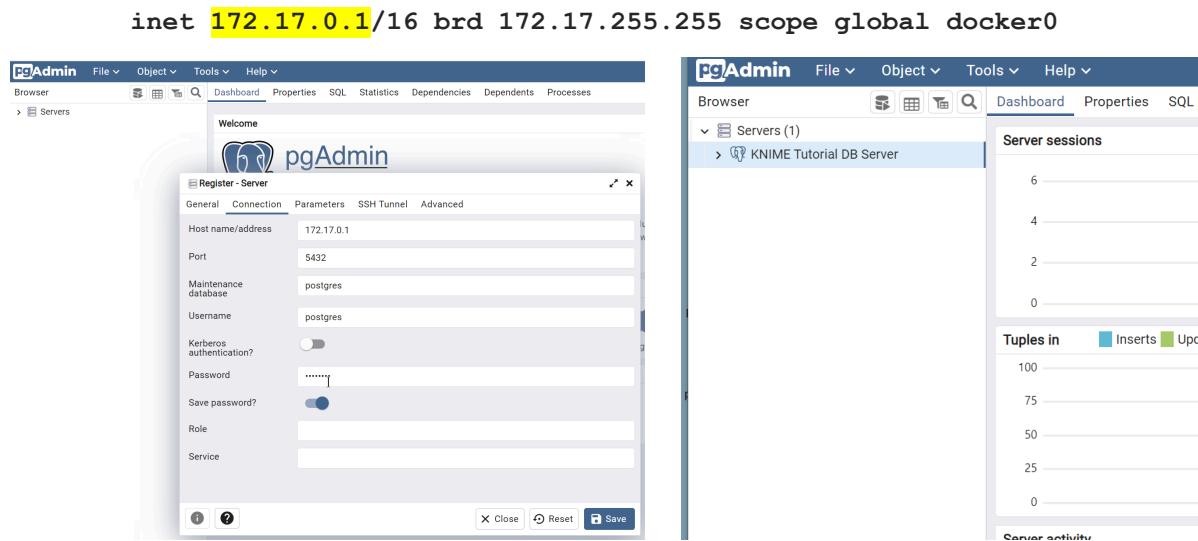
<https://job-gk0qpfj0kj2ybz63p36by5kj.dnanexus.cloud:8080>) with the specified credentials (user@domain.com, password).



To connect pgadmin in the container to the postgres database server port on the host, first obtain the docker0 interface IP address. This will be used in place of localhost in pgadmin (since localhost in pgadmin refers to the container local host). Add the workstation local database as a new server (data analysis workstation db) using the docker0 address (user *postgres*, password *password*).

```
ip addr show docker0

2: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    state UP group default
        link/ether 02:42:cd:c8:f1:0e brd ff:ff:ff:ff:ff:ff
```



Share Files Between Workstation FS and PostgreSQL Docker FS

Since pgadmin is running in a Docker container on the workstation, we are going to have to connect to the pgadmin container shell and copy files we want to share with pgadmin to the mount point shared by the container and the workstation (i.e. /home/dnanexus/db_backups). On a KNIME workstation terminal:

Connect to the shell in the pgadmin container.

```
sudo docker exec -it pgadmin sh
/pgadmin4 $
```

Copy files between the pgadmin backup directory to the container-host shared volume.

```
ls /var/lib/pgadmin/storage/user_domain.com
ls /home/dnanexus/db_backups
```

Control-D to exit the pgadmin container.

Add Shell and SQL Scripts for Use With KNIME

```
# Shell scripts for pfda cli upload-file, download, and ls
#
cd
pfda download -key $key --file-id file-GPf54j00Fk5xb2zgbKxV0JQ4-1
chmod ugo+x pfda-download-runner
sudo mv pfda-download-runner /usr/bin

pfda download -key $key --file-id file-GPgQZF00Fk5zxYX1QqY1v6XP-1
chmod ugo+x pfda-upload-runner
sudo mv pfda-upload-runner /usr/bin

pfda download -key $key --file-id file-GPf54j80Fk5x0BY71qvBB3Jf-1
chmod ugo+x pfda-ls-runner
sudo mv pfda-ls-runner /usr/bin

# Shell script for executing SQL from files using psql client
#
pfda download -key $key --file-id file-GPf54j00Fk5bVVK3BXK22p41-1
```

```

chmod ugo+x sql-runner
sudo mv sql-runner /usr/bin

# Shell script for ETL of data from csv.gz files into DB
#
pfda download -key $key --file-id file-GPgPGJ00Fk5q805K278f7V3G-1
chmod ugo+x EHR_Data_ETL.bash
sudo mv EHR_Data_ETL.bash /usr/bin

# DDL for tutorial DB
#
pfda download -key $key --file-id file-GPgKJ4j0Kj2k48B1yFGB117b-1
mv KNIME_Tutorial_EHR_Data_TableDDL_No2ndIndex.sql knime-workspace/

```

Download the KNIME Workflow

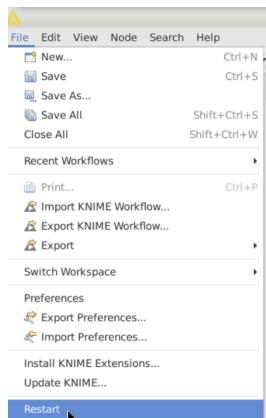
```

pfda download -key $key --file-id file-GPgy0bj0Fk5f7PGJf9vVJQPB-1
mv KNIME-Tutorial-20230217.knwf ~/knime-workspace/

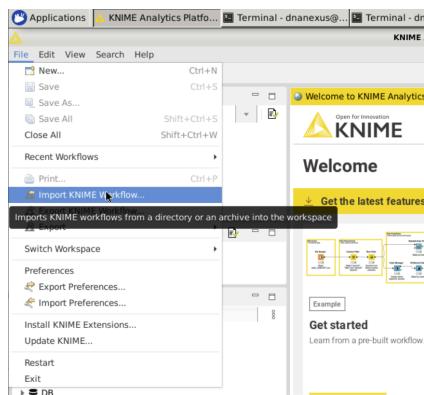
```

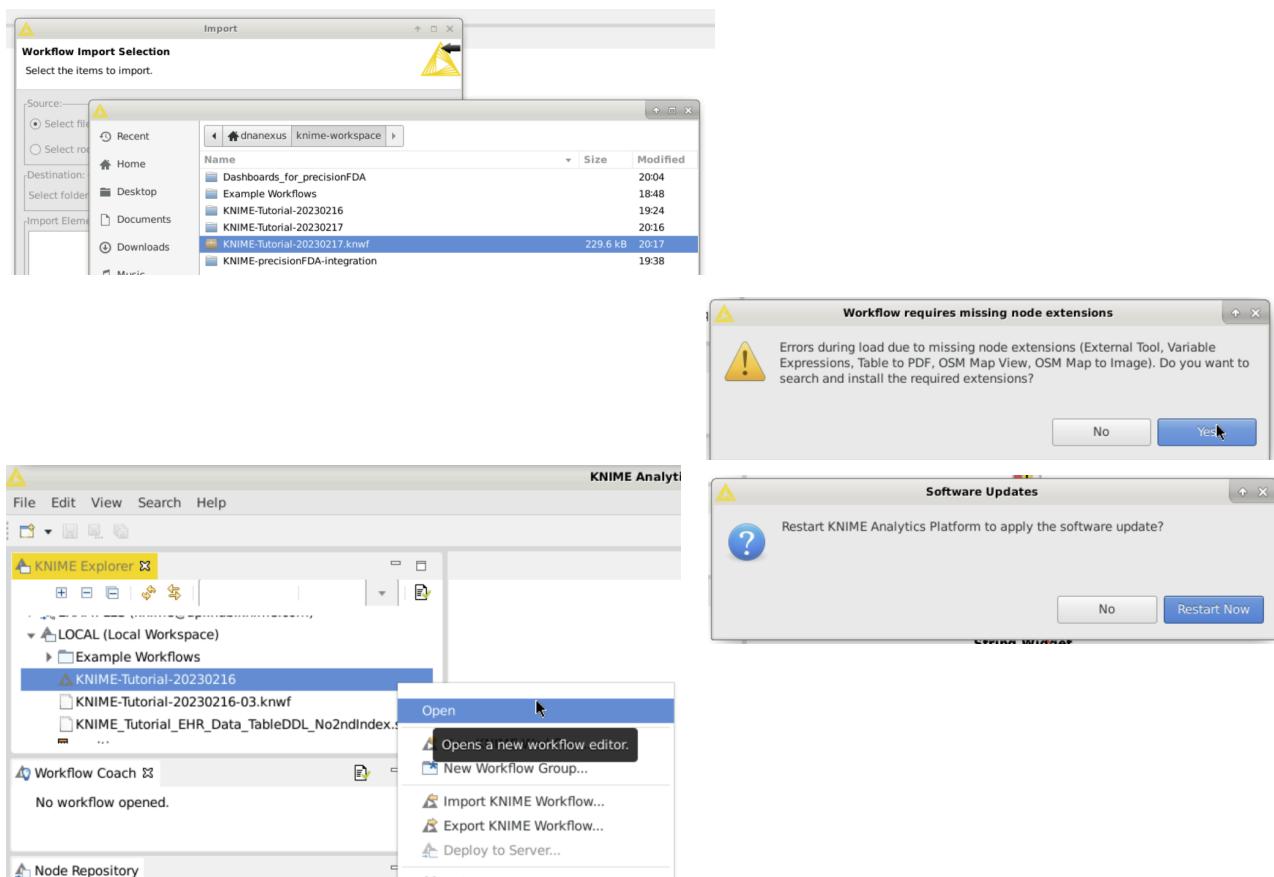
Run the KNIME Data Transformation Workflow

Restart KNIME to pickup the newly added files.

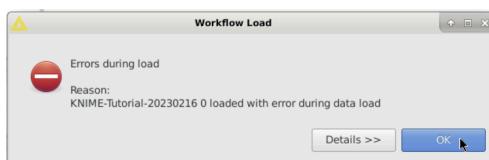


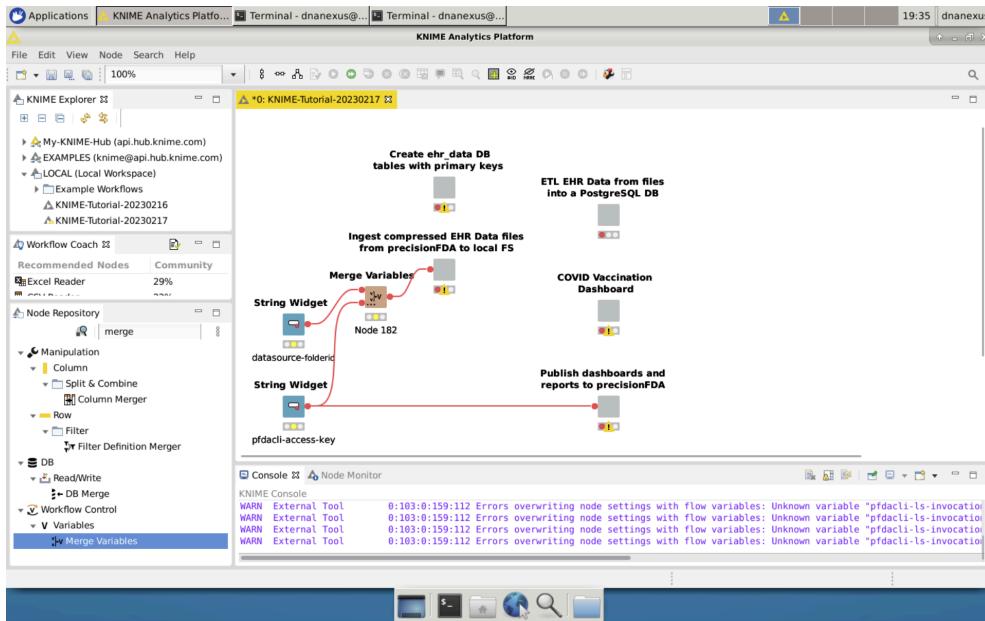
Import and Open the Workflow and Update Dependencies





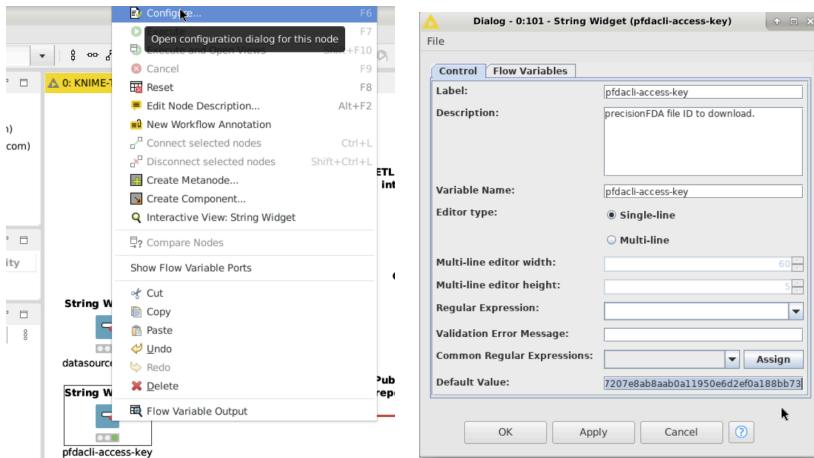
Ignore the warnings and errors.





Set the pFDA CLI Auth Token and Data Folder Variables

Configure the pfdacli-access-key String Widget to set the pfda CLI authentication token.



(Temporary workaround until the precisionFDA CLI is updated to properly perform ls on folders in the Everyone scope).

In precisionFDA, navigate to the KNIME Workstation Tutorial / Datafiles folder in the My Home Everyone context and select and download all six files to your local machine. Then, My Home / Files / Add Folder calling it “ KNIME sample data”, (or whatever you’d like since we’ll be referencing it by folder ID not name). Click into the new folder, and Add Files to re-upload the six files just downloaded. Copy the folder_id from the URL.

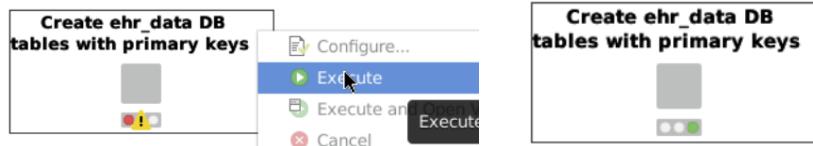
In KNIME, Configure the datasource-folderid String Widget to set the folder ID.

(Steps once the pFDA CLI is updated; ignore for now) Navigate to the KNIME Workstation Tutorial folder in the My Home Everyone context and copy the folder ID for the Datafiles folder from the browser URL. Configure the datasource-folderid String Widget to set the folder ID.

The screenshot shows the precisionFDA interface on the left and a KNIME dialog box on the right. The dialog box is titled 'Dialog - 0:115 - String Widget (datasource-folderid)'. It contains fields for 'Label' (datasource-folderid), 'Description' (precisionFDA datasource folder ID), 'Variable Name' (datasource-folderid), 'Editor type' (Single-line selected), 'Multi-line editor width' (60), 'Multi-line editor height' (10), 'Regular Expression' (dropdown menu), 'Validation Error Message' (empty), 'Common Regular Expressions' (dropdown menu), and 'Default Value' (8066400). Buttons at the bottom include OK, Apply, Cancel, and Help.

Create the DB

Execute the Create ehr_data DB tables with primary keys node.



Once the node shows green, refresh the KNIME Tutorial DB Server in pgAdmin to see the newly created knime_tutorial_ehr_data DB.

The pgAdmin interface shows the 'Server sessions' dashboard with a count of 6. Below it, the 'Databases' section lists 'knime_tutorial_ehr_data' and 'postgres'.

Download the Data from precisionFDA Folder

Execute the Ingest compressed EHR Data files from precisionFDA to local FS node.

Once the node shows green, check the downloaded files in the newly created EHR_Data directory.

```
dnanexus@job-GPgQfXQ0Kj2YB67yj00VJjVB:~$ tree EHR_Data/
EHR_Data/
├── ALLERGY
│   ├── ALLERGY_001.csv.gz
│   └── ALLERGY_002.csv.gz
├── ALLERGY_download.log
├── IMMUNIZATION
│   ├── IMMUNIZATION_001.csv.gz
│   └── IMMUNIZATION_002.csv.gz
├── IMMUNIZATION_download.log
└── PATIENT
    ├── PATIENT_001.csv.gz
    └── PATIENT_002.csv.gz
PATIENT_download.log

3 directories, 9 files
dnanexus@job-GPgQfXQ0Kj2YB67yj00VJjVB:~$
```

ETL the Data into the DB

Execute the ETL EHR Data from files into a PostgreSQL DB node to ETL the data from the compressed csv files into the DB.



Once the shows green, check the DB for content in pgadmin.

Analyze the Data and Create Reports

Execute the Dashboard node to and when it shows green, inspect the data table and geomap in the interactive node view.

The screenshot shows the KNIME interface with an 'Interactive View: Dashboard' selected. The dashboard displays a table with 10 entries, each consisting of a state abbreviation and a value. Below the table is a map of North America with green dots indicating data points. A context menu is open over the table, showing options like 'Interactive View: Dashboard', 'Show Flow Variable Ports', 'Cut', 'Copy', 'Paste', and 'Undo'.

□	■ AL	?	?	20
□	■ AR	20	20	?
□	■ AZ	?	?	20
□	■ CA	?	?	20
□	■ CO	20	20	?
□	■ CT	?	?	20
□	■ DE	20	20	?
□	■ FL	?	?	20
□	■ GA	20	20	?

Showing 1 to 10 of 50 entries

Previous 1 2 3 4 5 Next

KNIME on precisionFDA Tutorial

Map showing data points across North America. Labels include: Calgary, United States of America, Montreal, Toronto, Chicago, New York, Philadelphia, San Jose.

Reset Apply Close ▾

Publish the Reports to precisionFDA My Home

Execute the Publish dashboards and reports to precisionFDA node to upload the reports to your My Home files.



The screenshot shows the precisionFDA My Home interface. The top navigation bar includes links for Overview, Discussions, Challenges, Experts, My Home, Notes, Comparisons, Spaces, GSRS, Support, Get Started, and user profile information. The 'My Home' tab is active. The main content area shows a file list under the 'Files' category. The sidebar on the left provides a summary of other categories: Apps (21), Databases (11), Assets (4), Workflows (2), and Executions (338). The file list table has columns for Name, Size, Created, and Origin. Three files are listed:

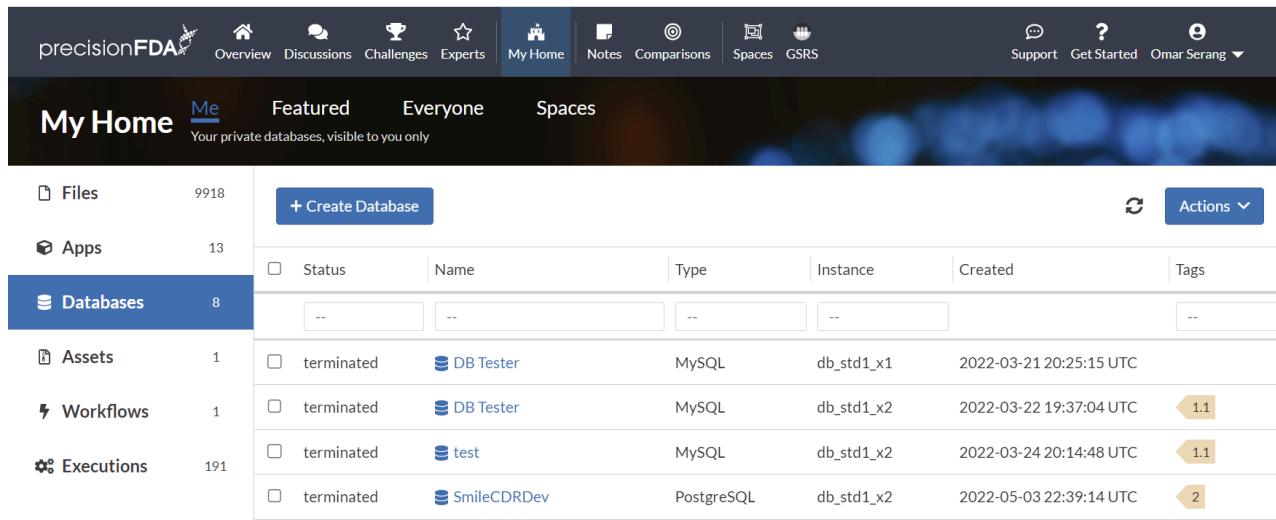
Name	Size	Created	Origin
KNIME-Tutorial-20230217.knwf	224 KB	2023-02-17 20:11:51 UTC	KNIME Tutorial
KNIME_on_pFDA_Tutorial_Geomap.png	96.6 KB	2023-02-17 20:09:18 UTC	KNIME Tutorial
KNIME_on_pFDA_Tutorial.pdf	4.22 KB	2023-02-17 20:09:16 UTC	KNIME Tutorial

Deploy a precisionFDA Database Cluster

PrecisionFDA provides AWS Aurora RDS database clusters that are accessible from Apps and Workstations. You will need to request DB Cluster access for your precisionFDA username in order to use this capability.

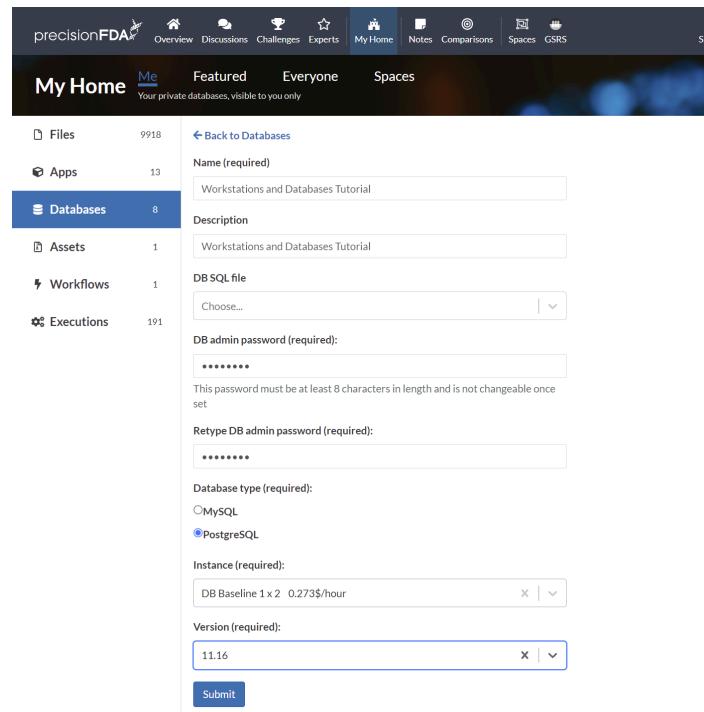
Create the Database

Select the Databases tab in My Home and click the Create Database button.



	Status	Name	Type	Instance	Created	Tags
--	--	--	--	--	--	--
<input type="checkbox"/> terminated		DB Tester	MySQL	db_std1_x1	2022-03-21 20:25:15 UTC	
<input type="checkbox"/> terminated		DB Tester	MySQL	db_std1_x2	2022-03-22 19:37:04 UTC	1.1
<input type="checkbox"/> terminated		test	MySQL	db_std1_x2	2022-03-24 20:14:48 UTC	1.1
<input type="checkbox"/> terminated		SmileCDRDev	PostgreSQL	db_std1_x2	2022-05-03 22:39:14 UTC	2

Create a “Workstations and Databases Tutorial” database, “password”, PostgreSQL 11.16 on the smallest available database instance type, and click the Submit button.



Name (required)
Workstations and Databases Tutorial

Description
Workstations and Databases Tutorial

DB SQL file
Choose...

DB admin password (required):

This password must be at least 8 characters in length and is not changeable once set

Retype DB admin password (required):

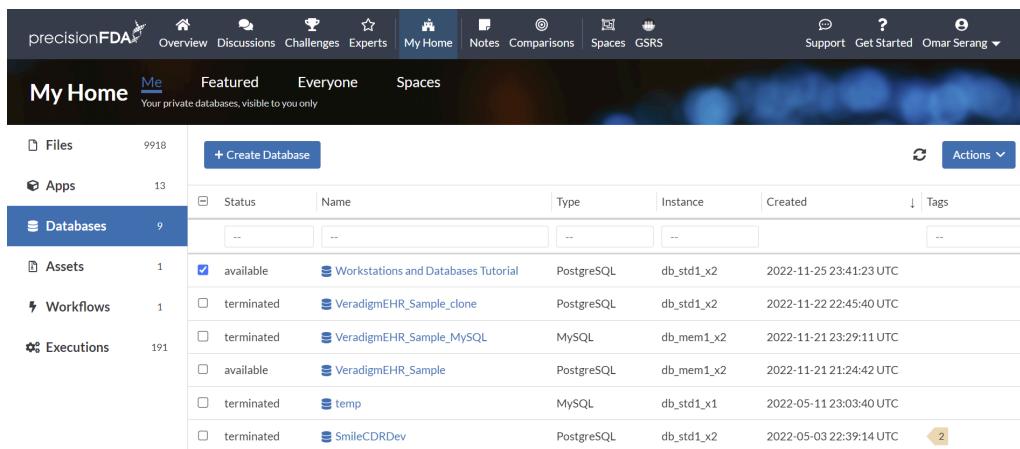
Database type (required):
 MySQL
 PostgreSQL

Instance (required):
DB Baseline 1 x 2 0.273\$/hour

Version (required):
11.16

Submit

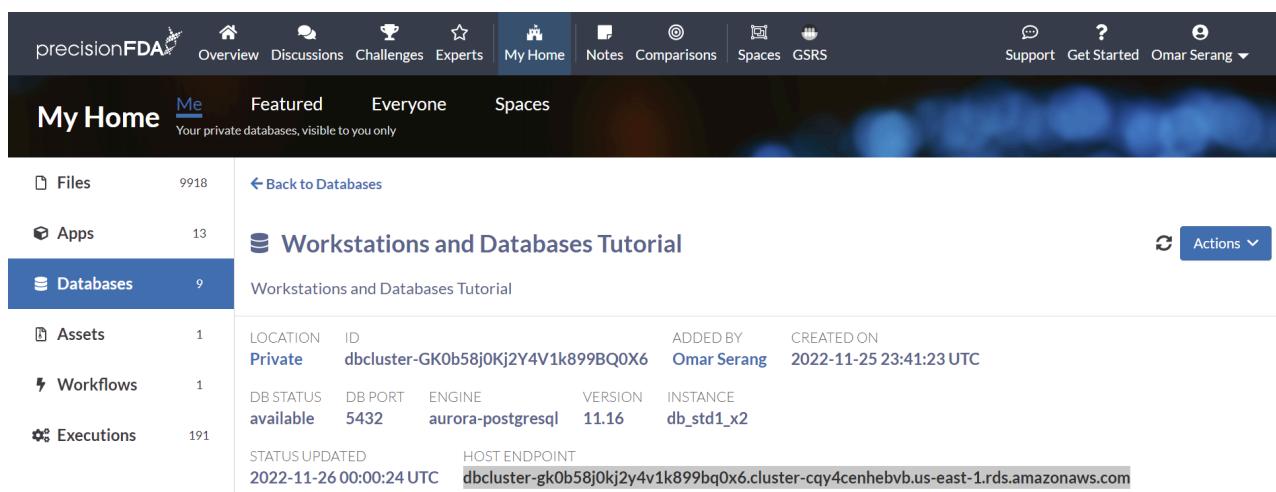
Refresh the database status using the  button until the database is available.



The screenshot shows the 'My Home' dashboard with the 'Databases' tab selected. A table lists several databases, including 'Workstations and Databases Tutorial' which is marked as 'available'. The table columns are Status, Name, Type, Instance, Created, and Tags.

	Status	Name	Type	Instance	Created	Tags
<input checked="" type="checkbox"/>	available	Workstations and Databases Tutorial	PostgreSQL	db_std1_x2	2022-11-25 23:41:23 UTC	
<input type="checkbox"/>	terminated	VeradigmEHR_Sample_clone	PostgreSQL	db_std1_x2	2022-11-22 22:45:40 UTC	
<input type="checkbox"/>	terminated	VeradigmEHR_Sample_MySQL	MySQL	db_mem1_x2	2022-11-21 23:29:11 UTC	
<input type="checkbox"/>	available	VeradigmEHR_Sample	PostgreSQL	db_mem1_x2	2022-11-21 21:24:42 UTC	
<input type="checkbox"/>	terminated	temp	MySQL	db_std1_x1	2022-05-11 23:03:40 UTC	
<input type="checkbox"/>	terminated	SmileCDRDev	PostgreSQL	db_std1_x2	2022-05-03 22:39:14 UTC	2

Click on the Workstations and Databases Tutorial database to open the detail page and copy the host endpoint URL.



The screenshot shows the 'My Home' dashboard with the 'Databases' tab selected. The 'Workstations and Databases Tutorial' database is selected, showing its details. The 'HOST ENDPOINT' field is highlighted.

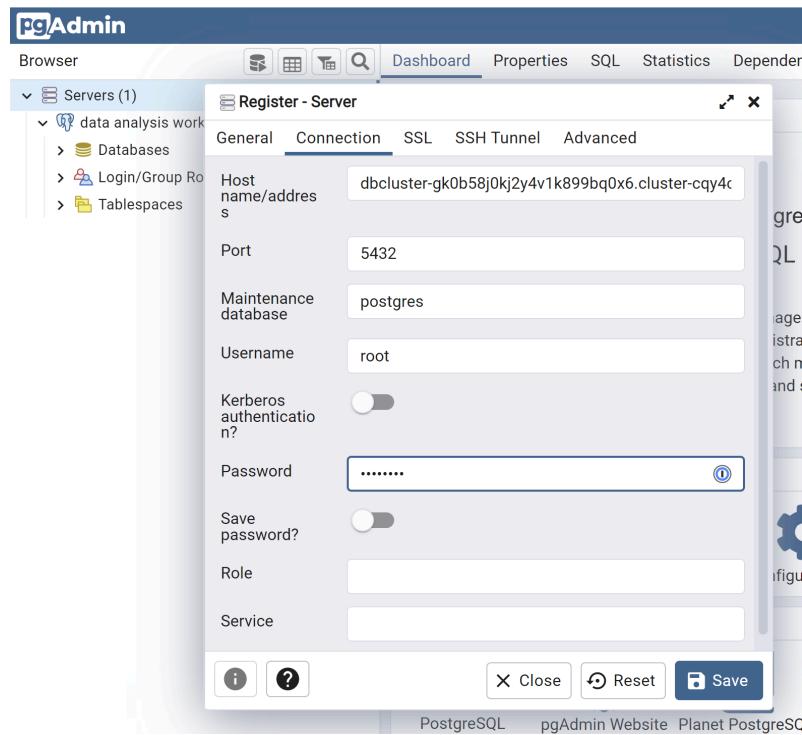
LOCATION	ID	ADDED BY	CREATED ON
Private	dbcluster-GK0b58j0Kj2Y4V1k899BQ0X6	Omar Serang	2022-11-25 23:41:23 UTC

DB STATUS	DB PORT	ENGINE	VERSION	INSTANCE
available	5432	aurora-postgresql	11.16	db_std1_x2

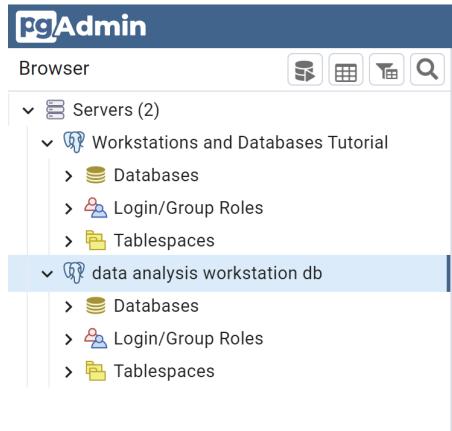
STATUS UPDATED	HOST ENDPOINT
2022-11-26 00:00:24 UTC	dbcluster-gk0b58j0kj2y4v1k899bq0x6.cluster-cqy4cenhebvb.us-east-1.rds.amazonaws.com

Connect to the cluster DB from pgadmin

In the pgadmin web service, add a new server for the Workstations and Databases Tutorial DB cluster using the host endpoint, user root, and the password specified when the database was created.



Note that we now have connections to both the local database on the data analysis workstation, and the cluster database.



Create a new database and tables

Connect to the cluster database from psql in the data analysis workstation shell.

```
PGPASSWORD="password" psql
--host=dbcluster-gbfqzqq0kj2jxgj109354vjj.cluster-cqy4cenhebvb.us-east-1.rds.amazonaws.com --username=root -d postgres
```

Using psql, create a new database.

```
-- Database: workstations_and_databases_tutorial_db
CREATE DATABASE workstations_and_databases_tutorial_db
    WITH
        OWNER = root
```

```

ENCODING = 'UTF8'
CONNECTION LIMIT = -1
IS_TEMPLATE = False;
    
```

Connect to the new database and create two tables.

```

\c workstations_and_databases_tutorial_db;

psql (9.5.25, server 11.16)
WARNING: psql major version 9.5, server major version 11.
          Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher:
ECDHE-RSA-AES128-GCM-SHA256, bits: 128, compression: off)
You are now connected to database
"workstations_and_databases_tutorial_db" as user "root".
workstations_and_databases_tutorial_db=>

CREATE TABLE public."PATIENT" (
    patient_id bigint NOT NULL,
    name character varying,
    gender character varying,
    zip character varying,
    country character varying,
    created_date date
);

CREATE TABLE public."OBSERVATION" (
    observation_id bigint NOT NULL,
    patient_id bigint,
    observation_name character varying,
    loinc character varying,
    created_date date
);

\dt
      List of relations
 Schema |     Name      | Type  | Owner
-----+-----+-----+-----
 public | OBSERVATION | table | root
 public | PATIENT    | table | root
(2 rows)
    
```

Load the cluster database from delimited text files

Although the workflow illustrated here may seem over-engineered for loading two data files, the techniques presented here were used to reliably and efficiently transfer tens of thousands of files and 15+ TB of data to precisionFDA.

In the data analysis workstation shell, create a datafiles directory

```
mkdir datafiles
```

Create and upload delimited data files

On your local client (i.e. laptop), create file **patients.txt** with the following content:

```

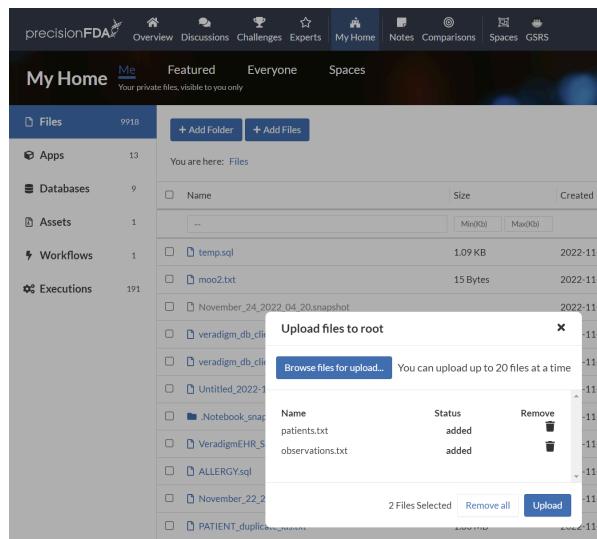
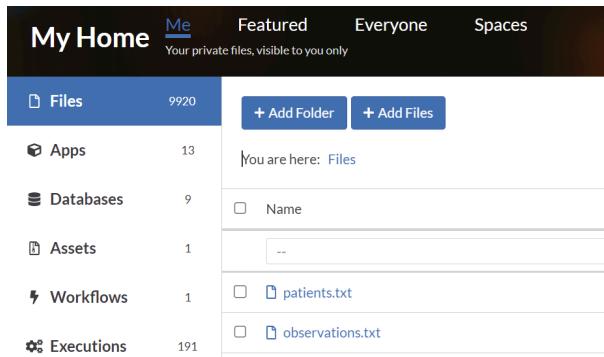
12345|Fred Foobar|M|94040|USA|2022-10-25
12346|Mary Merry|F|94040|USA|2022-09-24
    
```

12347|Barney Rubble|M|94040|USA|2022-08-23

Create file **observations.txt** with the following content:

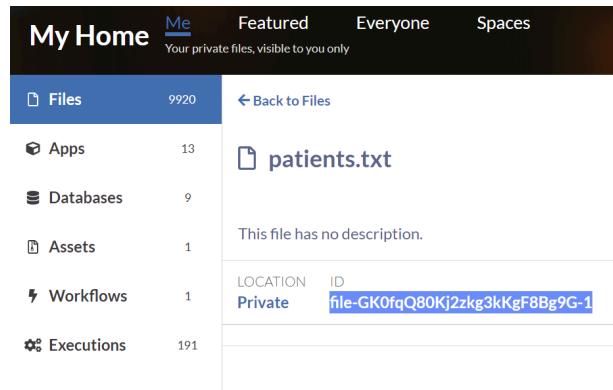
```
9870|12345|Annual check up|66678-4|2022-11-01
9871|12345|Emergency|LG32756-5|2022-11-02
9872|12346|Clinic visit|66678-4|2022-11-03
9873|12347|Lab results|74418-5|2022-11-04
9874|12347|Post-op checkup|65375-8|2022-11-05
```

In My Home / Files use the Add Files button to upload the two files to your private area.

Create and upload a manifest of data file IDs

Click into patients.txt and observations.txt details pages and copy their file IDs into a file named manifest.txt file on your local client.



The screenshot shows the 'My Home' dashboard with a sidebar on the left containing links for Files (9920), Apps (13), Databases (9), Assets (1), Workflows (1), and Executions (191). The main area displays the details for the 'patients.txt' file, which has no description. It shows the file is located in the Private space and has the ID 'file-GK0fqQ80Kj2zkg3kKgF8Bg9G-1'. A blue box highlights this ID.

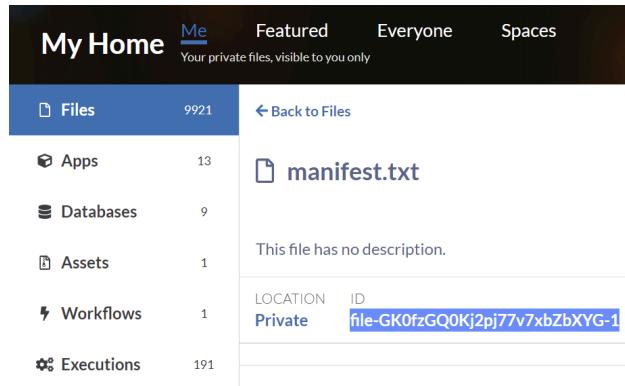
observations.txt

This file has no description.

LOCATION ID
Private [file-GK0fqGQ0Kj2gBZjxF24493PY-1](#)

```
-- manifest.txt
file-GK0fqQ80Kj2zkg3kKgF8Bg9G-1
file-GK0fqGQ0Kj2gBZjxF24493PY-1
```

Use the Add Files button to upload the `manifest.txt` file to your private area. Click into the details for the uploaded file and copy the file ID.



The screenshot shows the 'My Home' dashboard with a sidebar on the left containing links for Files (9921), Apps (13), Databases (9), Assets (1), Workflows (1), and Executions (191). The main area displays the details for the 'manifest.txt' file, which has no description. It shows the file is located in the Private space and has the ID 'file-GK0fzGQ0Kj2pj77v7xbZbXYG-1'. A blue box highlights this ID.

[Download the files in the manifest to the Data Analysis Workstation](#)

Using pfda CLI in the data analysis workstation shell, download the `manifest.txt` file to the workstation filesystem.

```
pfda download -file-id file-GK0fzGQ0Kj2pj77v7xbZbXYG-1
```

```
ls -l
-rw-r--r-- 1 root root 66 Nov 26 01:58 manifest.txt
```

Iterate through manifest and download data files

In the data analysis workstation shell install and run dos2unix on the manifest.txt file to ensure there are no cross-OS end-of-line issues.

```
cd/datafiles
apt install dos2unix
dos2unix manifest.txt

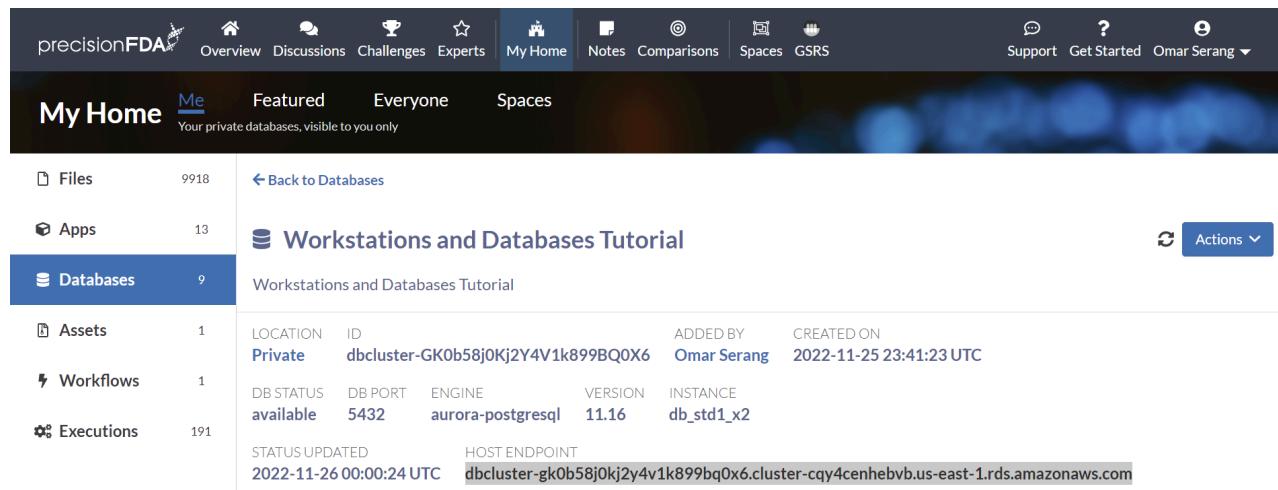
for FILE in $(cat manifest.txt); do pfda download -key $key -file-id
$FILE; done

ls
manifest.txt observations.txt patients.txt
```

Copy the data into the cluster DB tables

[Connect to the workstations_and_databases_tutorial_db cluster database](#)

Using the database host endpoint, connect to the workstations_and_databases_tutorial_db cluster database using psql on the data analysis workstation:



Workstations and Databases Tutorial					
Workstations and Databases Tutorial					
LOCATION	ID	ADDED BY	CREATED ON		
Private	dbcluster-GK0b58j0kj2y4v1k899BQ0X6	Omar Serang	2022-11-25 23:41:23 UTC		
DB STATUS	DB PORT	ENGINE	VERSION	INSTANCE	
available	5432	aurora-postgresql	11.16	db_std1_x2	
STATUS UPDATED	HOST ENDPOINT				
2022-11-26 00:00:24 UTC	dbcluster-gk0b58j0kj2y4v1k899bq0x6.cluster-cqy4cenhebvb.us-east-1.rds.amazonaws.com				

```
PGPASSWORD="password" psql
--host=dbcluster-gk0b58j0kj2y4v1k899bq0x6.cluster-cqy4cenhebvb.us-eas
t-1.rds.amazonaws.com --username=root -d
workstations_and_databases_tutorial_db

workstations_and_databases_tutorial_db=>
```

[Copy the patients and observations data into the cluster DB](#)

In psql:

```
\copy public."PATIENT" from '/home/dnanexus/datafiles/patients.txt'
delimiter '||' NULL ''

\copy public."OBSERVATION" from
'/home/dnanexus/datafiles/observations.txt' delimiter '||' NULL ''

select * from public."PATIENT";
```

```

patient_id |      name      | gender | zip   | country | created_date
-----+-----+-----+-----+-----+-----+
|
12345 | Fred Foobar    | M      | 94040 | USA     | 2022-10-25
12346 | Mary Merry     | F      | 94040 | USA     | 2022-09-24
12347 | Barney Rubble   | M      | 94040 | USA     | 2022-08-23

select * from public."OBSERVATION";
observation_id | patient_id | observation_name | loinc
|created_date
-----+-----+-----+-----+
|
9870 | 12345 | Annual check up | 66678-4 |
2022-11-01
9871 | 12345 | Emergency       | LG32756-5 |
2022-11-02
9872 | 12346 | Clinic visit     | 66678-4 |
2022-11-03
9873 | 12347 | Lab results       | 74418-5 |
2022-11-04
9874 | 12347 | Post-op checkup  | 65375-8 |
2022-11-05

```

Observe the new tables and data in the pgadmin Workstations and Databases Tutorial server connection.

	observation_id	patient_id	observation_name	loinc	created_date
1	9870	12345	Annual check up	66678-4	2022-11-01
2	9871	12345	Emergency	LG32756-5	2022-11-02
3	9872	12346	Clinic visit	66678-4	2022-11-03
4	9873	12347	Lab results	74418-5	2022-11-04
5	9874	12347	Post-op checkup	65375-8	2022-11-05

Connect RStudio to the cluster DB

In the RStudio console:

```

library(DBI)
con <- DBI::dbConnect(
  RPostgres::Postgres(),
  host =
"dbcluster-gk0b58j0kj2y4v1k899bq0x6.cluster-cqy4cenhebvb.us-east-1.rds.amazonaws.com",

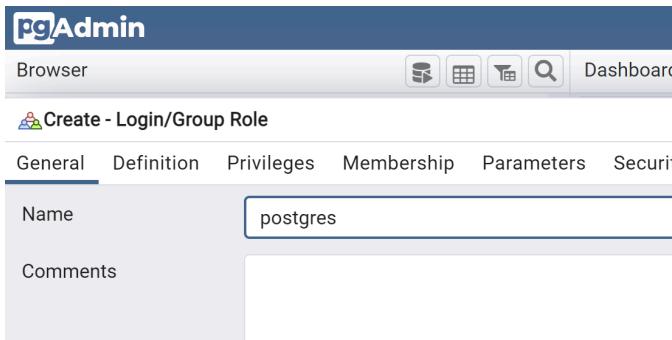
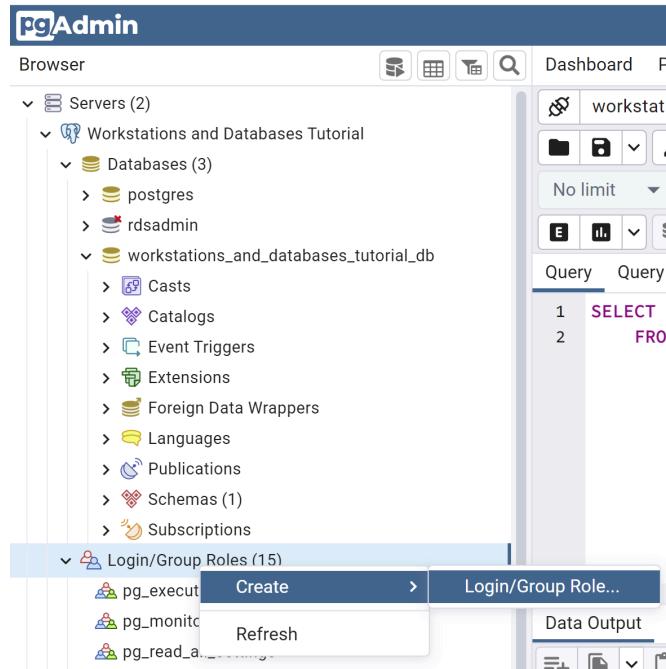
```

```
    port = 5432, dbname = "workstations_and_databases_tutorial_db",
    user = "root", password = "password"
)
dbListTables(con)
[1] "OBSERVATION" "PATIENT"
```

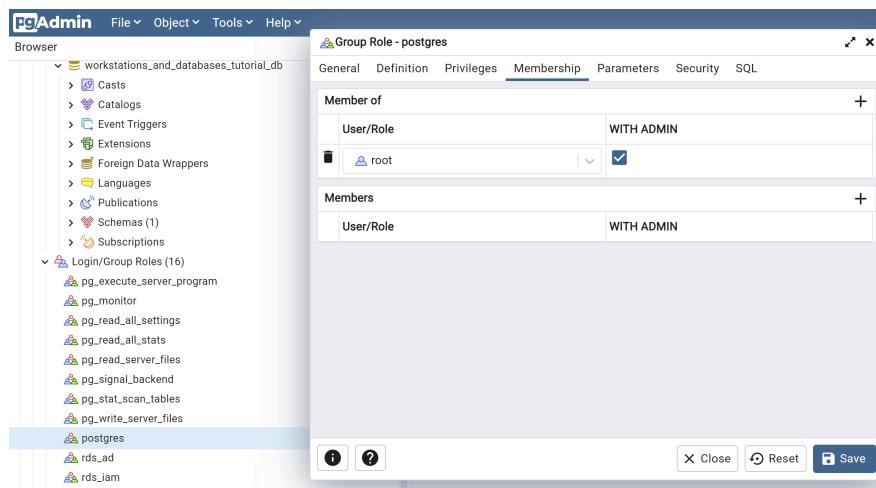
Backup the cluster DB and restore it to local DBs

Add a postgres role to the cluster DB

Right-click Login/Group Roles in the Workstations and Databases Tutorial server connection in pgadmin and add a postgres role.



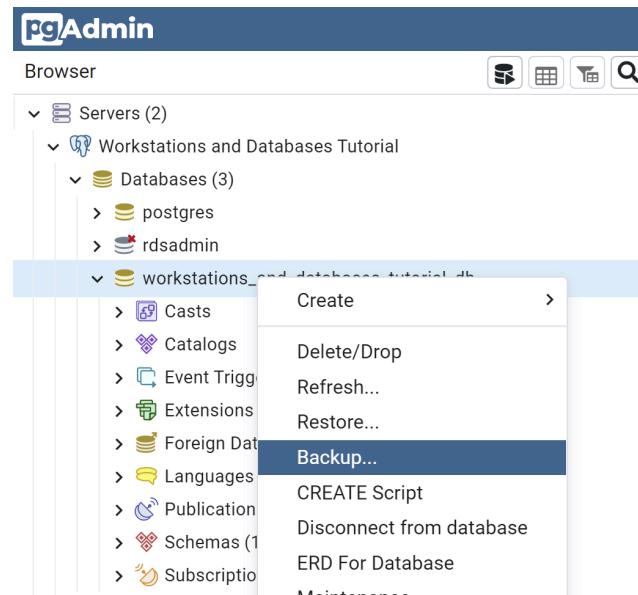
Right-click the postgres role and select properties, and add the to the root group with admin privileges in the Membership tab.



The screenshot shows the pgAdmin interface with the 'Group Role - postgres' configuration dialog open. The 'Membership' tab is selected, displaying a table with one row: 'User/Role' root and 'Privileges' WITH ADMIN. There is a checkmark next to 'WITH ADMIN'. Below this is a 'Members' section which is currently empty.

Backup the cluster DB using pgadmin

Select the workstations_and_databasesTutorial_db database in the Workstations and Databases Tutorial server connection in pgadmin and right-click to backup the database.



The screenshot shows the pgAdmin interface with the 'workstations_and_databasesTutorial_db' database selected in the tree view. A context menu is open over the database, with the 'Backup...' option highlighted in blue. Other options in the menu include 'Create', 'Delete/Drop', 'Refresh...', 'Restore...', 'CREATE Script', 'Disconnect from database', and 'ERD For Database'.

Specify a backup filename (e.g. workstations_and_databasesTutorial_db-2022-11-25.tar), format as Tar, assign role name *postgres* and set all the Data/Objects Do not save options..

pgAdmin

Backup (Database: workstations_and_databasesTutorial_db)

- General Data/Objects Options

Filename: workstations_and_databasesTutorial_db-2022-11-25.tar

Format: Tar

Compression ratio:

Encoding: Select an item...

Number of jobs:

Role name: postgres

pgAdmin

Backup (Database: workstations_and_databasesTutorial_db)

- General Data/Objects Options

Type of objects:

- Only data:
- Only schema:
- Blobs:

Do not save:

- Owner:
- Privilege:
- Tablespace:
- Unlogged table data:
- Comments:

Process completed

Backing up an object on the server 'Workstations and Databases Tutorial (dbcluster-gk0b58j0kj2y4v1k899bq0x6.cluster-cqy4cenhebvb.us-east-1.rds.amazonaws.com:5432)' from database 'workstations_and_databasesTutorial_db'

[View Processes](#)

Process started

Backing up an object on the server 'Workstations and Databases Tutorial (dbcluster-gk0b58j0kj2y4v1k899bq0x6.cluster-cqy4cenhebvb.us-east-1.rds.amazonaws.com:5432)' from database 'workstations_and_databasesTutorial_db'

[View Processes](#)

Copy the backup file from the pgadmin container to the workstation filesystem

Since pgadmin is running in a Docker container on the data analysis workstation, we are going to have to connect to the pgadmin container shell and copy the backup file to the mount point shared by the container and the workstation (i.e. /home/dnanexus/db_backups). On the data analysis workstation:

Connect to the shell in the pgadmin container.

```
docker exec -it pgadmin sh
/pgadmin4 $
```

Copy the backup file from the pgadmin backup directory to the container-host shared volume.

```
cd /var/lib/pgadmin/storage/user_domain.com
workstations_and_databases_tutorial_db-2022-11-25

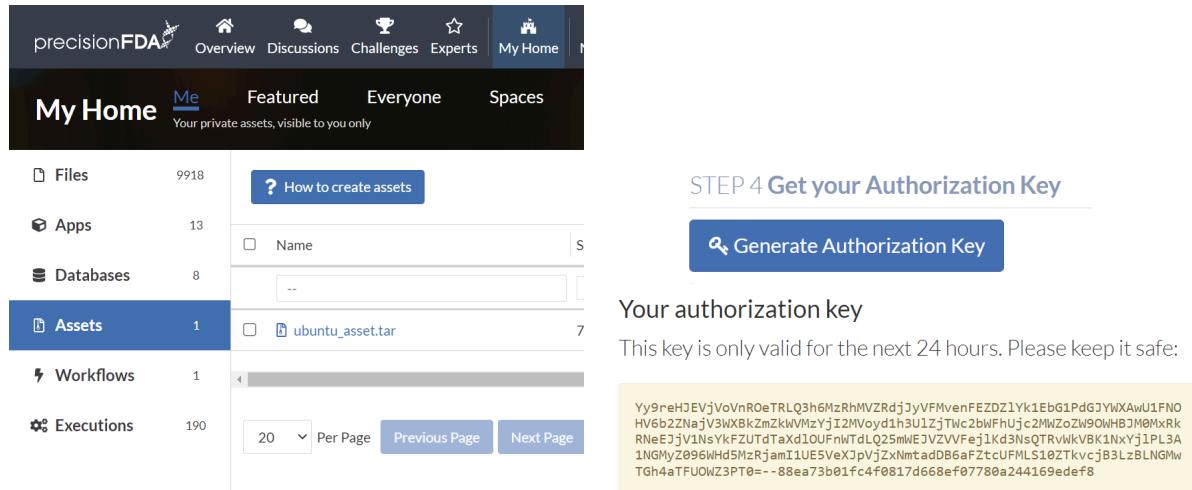
cp
/var/lib/pgadmin/storage/user_domain.com/workstations_and_databases_tutorial_db-2022-11-25.tar /home/dnanexus/db_backups
```

Control-D to exit the container shell and verify the presence of the backup file on the workstation in the container-host shared mount point.

```
ls db_backups/
workstations_and_databases_tutorial_db-2022-11-25
```

Upload the backup file to precisionFDA

Under My Home Assets, click on the How to create assets button to find the button to generate the temporary authorization key that you'll use with the CLI.

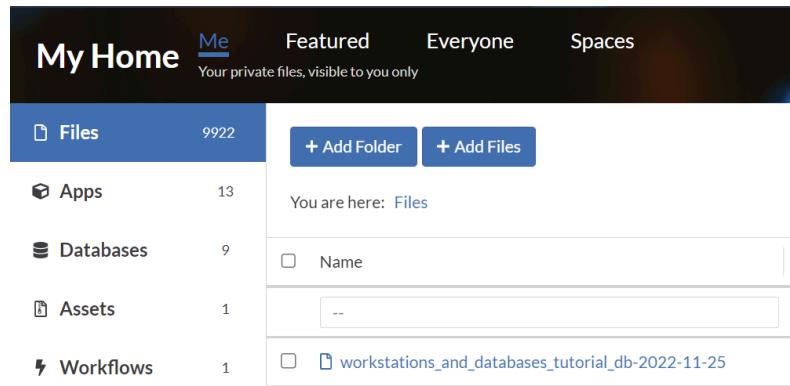


The screenshot shows the precisionFDA 'My Home' dashboard. In the sidebar, 'Assets' is selected, showing 1 item: 'ubuntu_asset.tar'. A modal window titled 'STEP 4 Get your Authorization Key' is open, containing a 'Generate Authorization Key' button and a large text area displaying a long string of characters representing the authorization key.

On the data analysis workstation shell:

```
key="..."

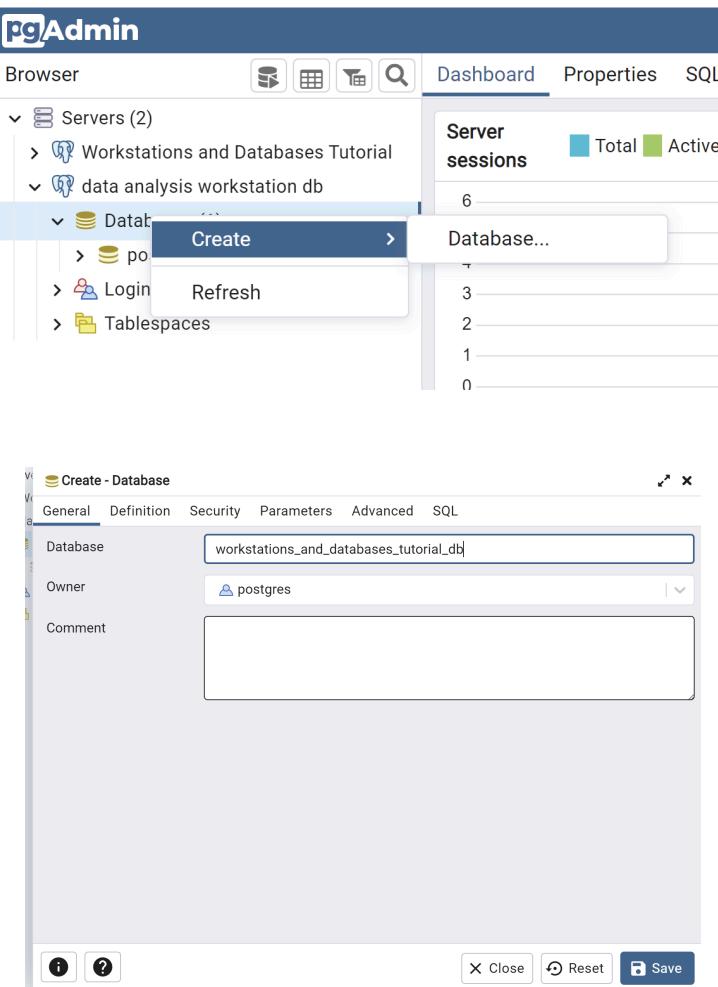
pfda upload-file -key $key -file
~/db_backups/workstations_and_databases_tutorial_db-2022-11-25.tar
```



The screenshot shows the 'My Home' dashboard of the precisionFDA platform. The top navigation bar includes 'Me', 'Featured', 'Everyone', and 'Spaces'. Below the navigation is a message: 'Your private files, visible to you only'. The main content area is titled 'Files' with a count of 9922. It also lists 'Apps' (13), 'Databases' (9), 'Assets' (1), and 'Workflows' (1). On the right, there are buttons for '+ Add Folder' and '+ Add Files'. A search bar indicates the user is looking for 'workstations_and_databases_tutorial_db-2022-11-25'.

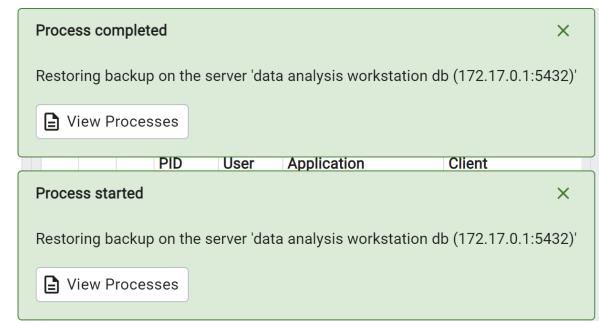
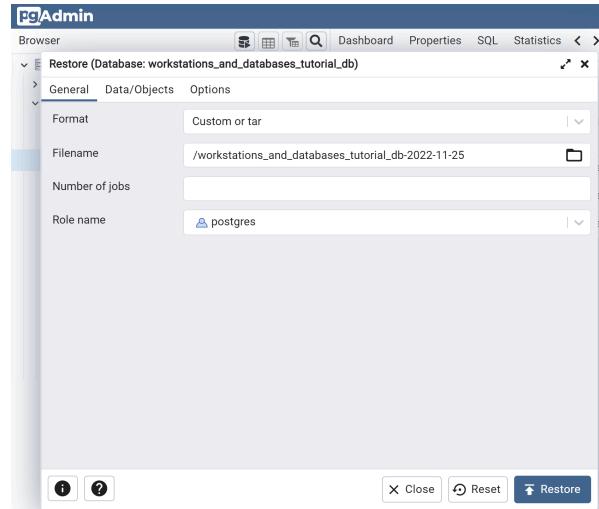
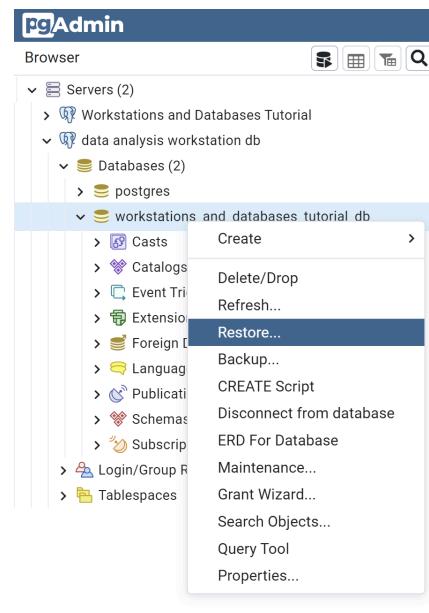
Restore the backup to the data analysis workstation local DB

Using the pgadmin connection to the data analysis workstation db, create a new database `workstations_and_databases_tutorial_db`, owner `postgres`.



The screenshot shows the pgAdmin interface. The left sidebar shows a tree view of servers, with 'data analysis workstation db' selected. A context menu is open over the 'Data' folder, with 'Create' highlighted. To the right, a 'Server sessions' panel shows 6 total sessions. Below it, a 'Create - Database' dialog box is open. The 'General' tab is selected, showing 'Database' set to 'workstations_and_databases_tutorial_db' and 'Owner' set to 'postgres'. Other tabs include 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. At the bottom of the dialog are buttons for 'Close', 'Reset', and 'Save'.

Right-click on the new database on the data analysis and workstation db server connection and restore the backup to the local server (from the file in the pgadmin container), using custom or tar format, and the postgres role name.



Select the contents of the restored PATIENT and OBSERVATION tables.

The screenshot shows the pgAdmin interface. On the left, there's a tree view of database objects under 'workstations_and_databases_tutorial_db/postgres'. A query window at the top contains the SQL command: 'SELECT observation_id, patient_id, observation_name FROM public.OBSERVATION;'. Below it, a data output window shows a table with five rows of data:

	observation_id	patient_id	observation_name	locn	created_date
1	9870	12345	Annual check up	66678-4	2022-11-01
2	9871	12345	Emergency	LG32756-5	2022-11-02
3	9872	12346	Clinic visit	66678-4	2022-11-03
4	9873	12347	Lab results	74418-5	2022-11-04
5	9874	12347	Post-op checkup	65375-8	2022-11-05

Restore the backup to the data analysis notebook local DB

Under My Home Assets, click on the How to create assets button to find the button to generate the temporary authorization key that you'll use with the CLI.

The screenshot shows the 'Assets' section of the 'My Home' page. It lists one asset named 'ubuntu.asset.tar'. A blue button labeled 'How to create assets' is visible above the asset list. Below the asset list are buttons for 'Name' and 'Delete'.

STEP 4 Get your Authorization Key

[Generate Authorization Key](#)

Your authorization key

This key is only valid for the next 24 hours. Please keep it safe:

```
Yy9reHJEVjVoVnROeTRLQ3h6MzRhMVZrdjJyVFMvenFEZDZ1Yk1EbG1PdGJYWXAuW1FNO
HV6b2ZNaJv3WXBkZm2kVnMzYjI2MVooy1h3U1ZjTwc2bWFhUjC2MwZoZw90WHBjM0MxrK
RNcEJjV1NsYkFZUTdtaxd1OUFnTdTlQ25mWEJjVZVVFej1kd1d3NsQTRVwkvBK1NxYj1PL3A
1NGMyZ896Whd5MzRjamI1UE5VeXjPvJzxNmtdaDB6aFZtcUFMLS10ZTkvcjB3LzBLNGMw
TGh4aTFUOWZ3PT0=-88ea7b01fc4f0817d668ef07780a244169edef8
```

Click into the detail page for the backup file and copy the file ID.

The screenshot shows the detail page for the 'ubuntu.asset.tar' file. At the top, there's a 'Back to Files' link. The file details are shown below:

workstations_and_databases_tutorial_db-2022-11-25

This file has no description.

LOCATION	ID	ADDED BY	ORIGIN	FILE SIZE
Private	file-GK0k9x00Kj2vFJ67FB11z1qp-1	Omar Serang	Uploaded	9.5 KB

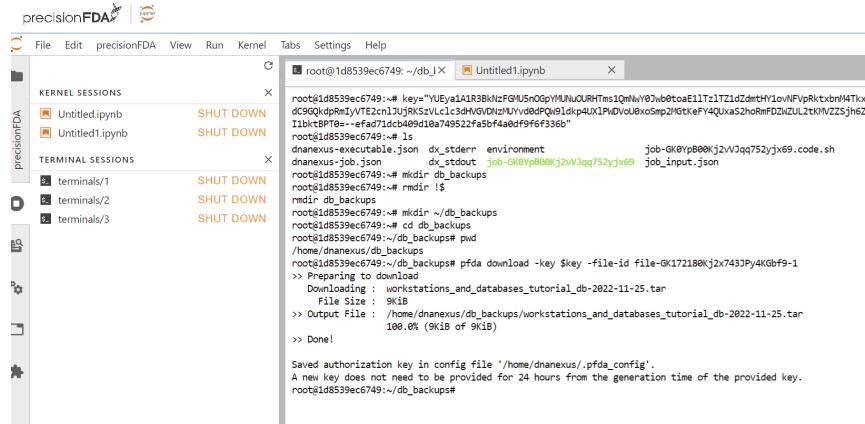
In a terminal window in the data analysis jupyterLab notebook, download the backup file using its file ID as copied in the step above:

```
mkdir ~/db_backups
```

50

20231122

```
cd db_backups  
key="..."  
pfda download -key $key -file-id file-GK172180Kj2x743JPy4KGbf9-1
```



In psql connected to the local host, create a new database `workstations_and_databases_tutorial_db`, and a new user `root`.

```
psql -U postgres -h 127.0.0.1
psql (15.1 (Ubuntu 15.1-1.pgdg18.04+1))
postgres=#
```

```
CREATE USER root;
```

```
CREATE DATABASE workstations_and_databases_tutorial_db
  WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    CONNECTION LIMIT = -1
    IS TEMPLATE = False;
```

Ctrl-D to exit psql and use restore the database from the backup file.

```
pg_restore --dbname=workstations_and_databases_tutorial_db --verbose  
~/db_backups/workstations_and_databases_tutorial_db-2022-11-25.tar -U  
postgres
```

You can ignore the errors associated with the root role not existing and use the Python notebook to select the contents from the restored database. We can observe the same results from newly restored database as from the cluster database that was the backup source. In a notebook Python code block:

```
import psycopg2
conn =
psycopg2.connect("dbname='workstations_and_databases_tutorial_db'
user='postgres' host='127.0.0.1'")
cur = conn.cursor()
cur.execute('SELECT * FROM public."PATIENT" limit 10')
# fetch results
rows = cur.fetchall()
# iterate through results
for row in rows:
    print ("PATIENT", row[0], row[1], row[2])
```

```

cur.execute('SELECT * FROM public."OBSERVATION" limit 10')
# fetch results
rows = cur.fetchall()
# iterate through results
for row in rows:
    print ("OBSERVATION", row[0], row[1], row[2])
    
```

```

[1]: import psycopg2
conn = psycopg2.connect("dbname='workstations_and_databases_tutorial_db' user='postgres' host='127.0.0.1'")
cur = conn.cursor()
cur.execute('SELECT * FROM public."PATIENT" limit 10')
# fetch results
rows = cur.fetchall()
# iterate through results
for row in rows:
    print ("PATIENT", row[0], row[1], row[2])
    print ("OBSERVATION", row[0], row[1], row[2])

PATIENT 12345 Fred Frobber M
PATIENT 12346 Harry Harry F
PATIENT 12347 Linda Linda M
OBSERVATION 9878 12345 Annual check up
OBSERVATION 9878 12346 Emergency room visit
OBSERVATION 9878 12346 Clinical visit
OBSERVATION 9878 12347 Lab results
OBSERVATION 9878 12347 Post-op checkup

[1]: conn = psycopg2.connect("dbname='workstations_and_databases_tutorial_db' user='root' host='dbcluster-gi40858j4k3j4:i48990ghs6.clust
[1]: cur = conn.cursor()
[1]: cur.execute('SELECT * FROM public."PATIENT" limit 10')
[1]: # fetch results
[1]: rows = cur.fetchall()
[1]: # iterate through results
[1]: for row in rows:
[1]:     print ("PATIENT", row[0], row[1], row[2])
[1]:     print ("OBSERVATION", row[0], row[1], row[2])

PATIENT 12345 Fred Frobber M
PATIENT 12346 Harry Harry F
PATIENT 12347 Linda Linda M
OBSERVATION 9878 12345 Annual check up
OBSERVATION 9878 12346 Emergency room visit
OBSERVATION 9878 12346 Clinical visit
OBSERVATION 9878 12347 Lab results
OBSERVATION 9878 12347 Post-op checkup
    
```

Stop or Terminate the Database Cluster

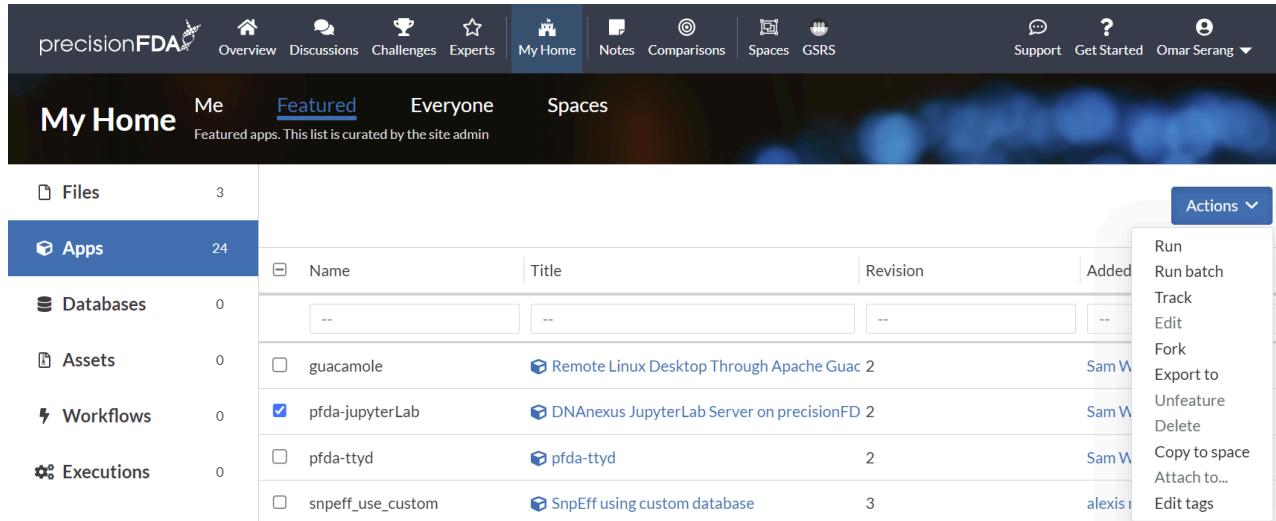
In My Home / Databases, select the database for action and either Stop or Terminate the database using the Action dropdown menu. If your data is already stored on precisionFDA and can be readily reconstituted into a new database, then select Terminate. If your database is a work in progress and you'd like to keep it intact while not using it overnight, or the weekend, then select Stop.

Status	Name	Type	Instance	Created	Tags
terminated	DB Tester	MySQL	db_std1_x1	2022-03-21 20:25:15 UTC	
terminated	DB Tester	MySQL	db_std1_x2	2022-03-22 19:37:04 UTC	11
terminated	test	MySQL	db_std1_x2	2022-03-24 20:14:48 UTC	11
terminated	SmileCDRDev	PostgreSQL	db_std1_x2	2022-05-03 22:39:14 UTC	2
terminated	temp	MySQL	db_std1_x1	2022-05-11 23:03:40 UTC	
stopping	VeradigmEHR_Sample	PostgreSQL	db_mem1_x2	2022-11-21 21:24:42 UTC	
terminated	VeradigmEHR_Sample_SQL	MySQL	db_mem1_x2	2022-11-21 23:29:11 UTC	
terminated	VeradigmEHR_Sample_clone	PostgreSQL	db_std1_x2	2022-11-22 22:45:40 UTC	
available	Workstations and Databases Tutorial	PostgreSQL	db_std1_x2	2022-11-25 23:41:23 UTC	

Build Data Analysis Notebook

Run the pfda-jupyterLab Featured App

Using the smallest instance type, run the Data Analysis Notebook job specifying PYTHON_R.

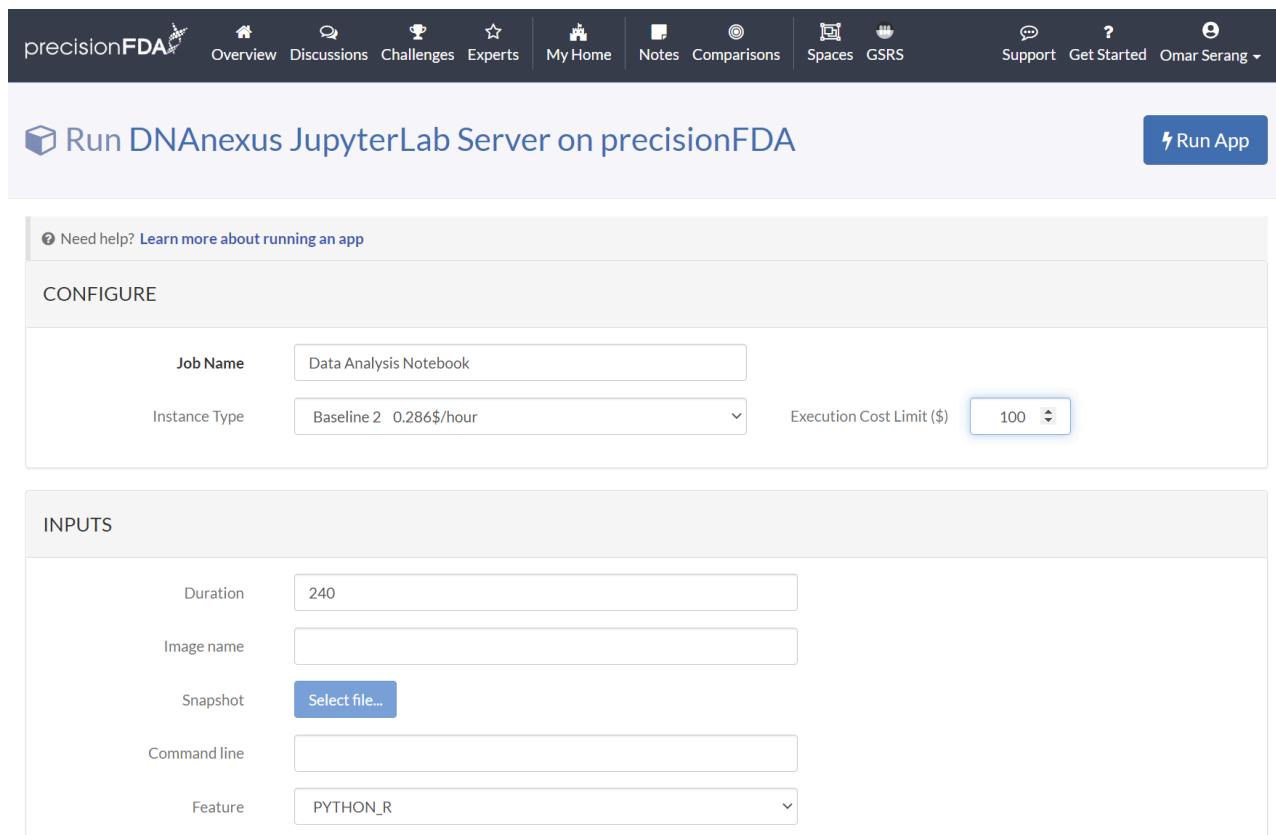


My Home Me Featured Everyone Spaces

Featured apps. This list is curated by the site admin

	Files	3	
	Apps	24	Actions ▾
	Databases	0	
	Assets	0	
	Workflows	0	
	Executions	0	

Name	Title	Revision	Added
--	--	--	--
guacamole	Remote Linux Desktop Through Apache Guac 2	Sam W	
<input checked="" type="checkbox"/> pfda-jupyterLab	DNAexus JupyterLab Server on precisionFDA 2	Sam W	
<input type="checkbox"/> pfda-ttyd	pfda-ttyd	2	Sam W
<input type="checkbox"/> snpeff_use_custom	SnpEff using custom database	3	alexis i



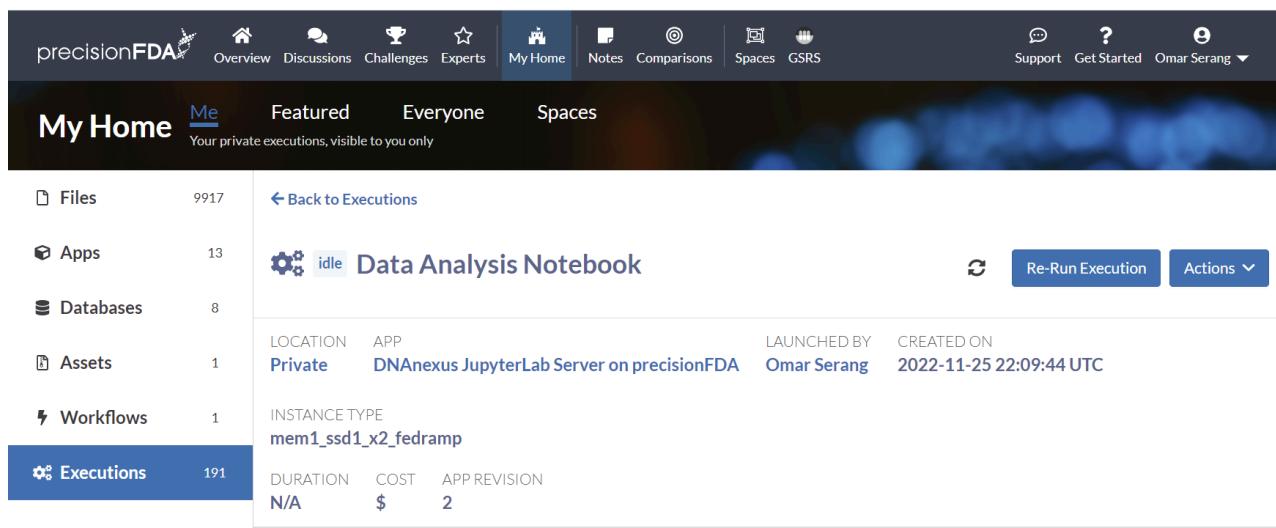
Need help? [Learn more about running an app](#)

CONFIGURE

Job Name	Data Analysis Notebook
Instance Type	Baseline 2 0.286\$/hour
Execution Cost Limit (\$)	100

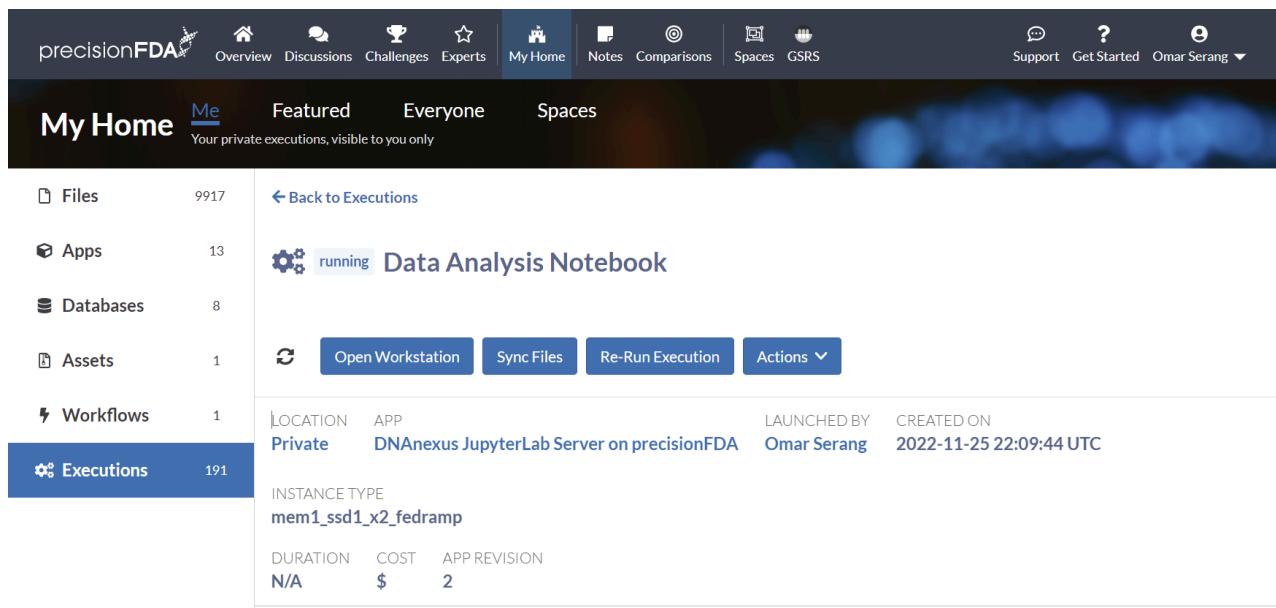
INPUTS

Duration	240
Image name	
Snapshot	Select file...
Command line	
Feature	PYTHON_R

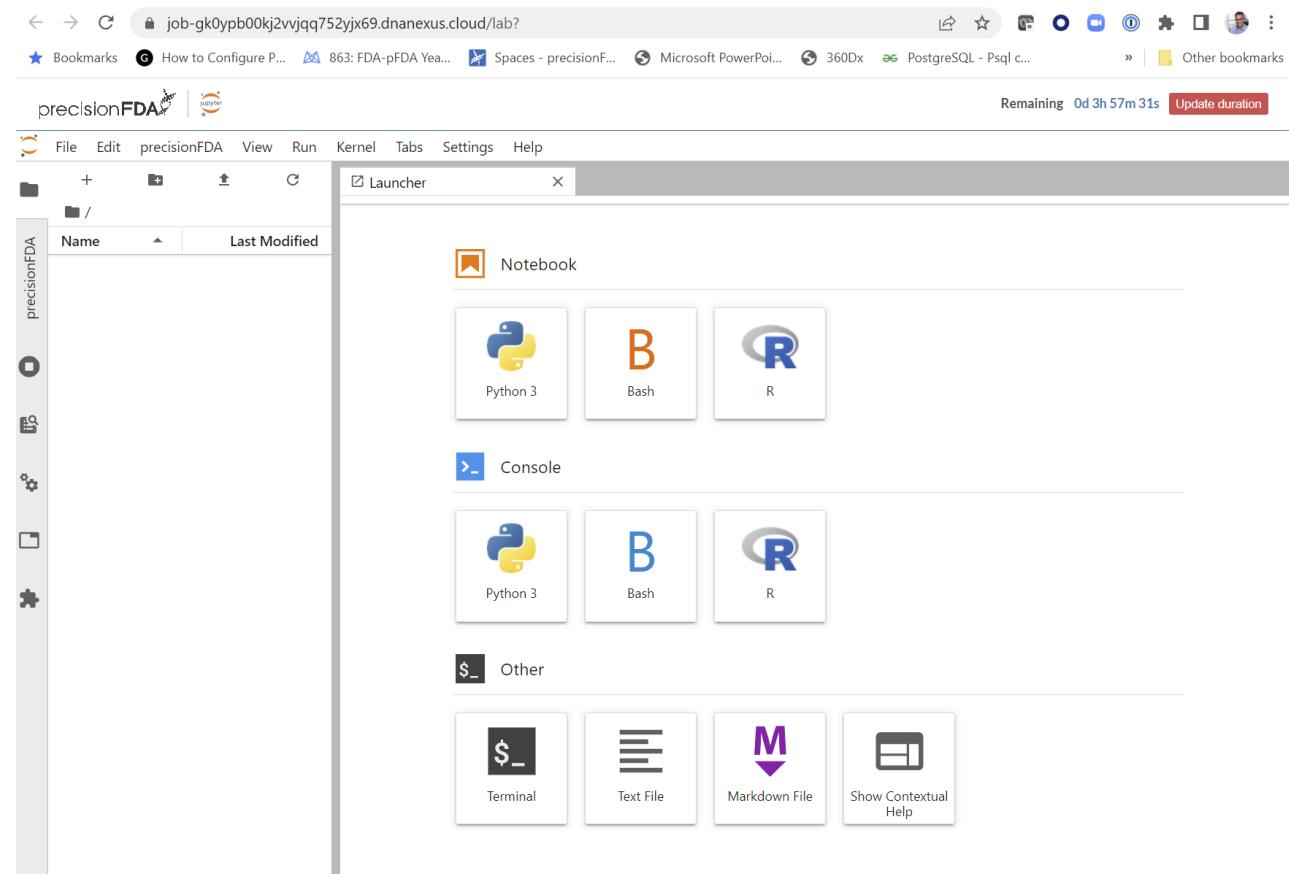


The screenshot shows the precisionFDA interface. At the top, there's a navigation bar with links like Overview, Discussions, Challenges, Experts, My Home, Notes, Comparisons, Spaces, and GSRS. Below that is a secondary navigation bar with tabs for My Home, Me, Featured, Everyone, and Spaces. A message says "Your private executions, visible to you only". On the left, a sidebar lists Files (9917), Apps (13), Databases (8), Assets (1), Workflows (1), and Executions (191). The main content area displays a "Data Analysis Notebook" execution. It shows the location is Private, the app is DNAnexus JupyterLab Server on precisionFDA, launched by Omar Serang on 2022-11-25 22:09:44 UTC, and the instance type is mem1_ssd1_x2_fedramp. There are buttons for Re-Run Execution and Actions.

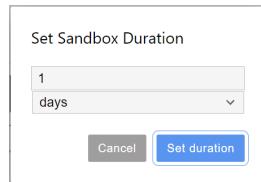
Refresh the execution status using the  button until the job is running and open the workstation. It may take a few minutes after the job is running for the notebook to open.



This screenshot shows the same interface after refreshing. The execution status has changed from "idle" to "running". The main content area now includes buttons for Open Workstation, Sync Files, Re-Run Execution, and Actions. The rest of the information (location, app, launched by, created on, instance type, duration, cost, app revision) remains the same.

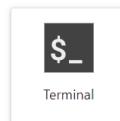


Adjust the remaining time-to-live for the notebook using the Update duration button.



Download and Install the pfda CLI

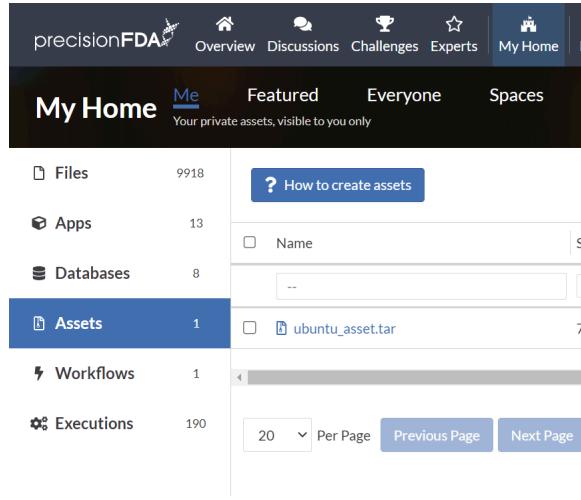
Copy the link for the current version of the Linux pFDA CLI from the CLI Docs page <https://precision.fda.gov/docs/cli>. Open a Terminal in the Data Analysis notebook and download and unpack the CLI.



```
-- Install pfda CLI
wget
https://pfda-production-static-files.s3.amazonaws.com/cli/pfda-linux-
2.1.2.tar.gz
tar xf pfda-linux-2.1.2.tar.gz
mv pfda /usr/bin/
```

```
pfda --version
```

Retrieve a CLI authorization key. Under My Home Assets, click on the How to create assets button to find links to the precisionFDA CLI, and the button to generate the temporary authorization key that you'll use with the CLI.



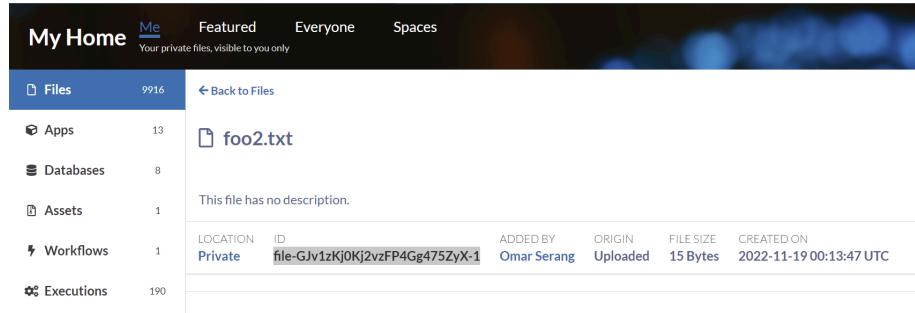
The screenshot shows the 'My Home' dashboard with various categories: Files (9918), Apps (13), Databases (8), Assets (1), Workflows (1), and Executions (190). The 'Assets' section is highlighted. A blue button labeled '? How to create assets' is positioned above the asset list. The asset list shows one item: 'ubuntu_asset.tar'.

STEP 3 Download the precisionFDA CLI

[Linux](#) [Mac OS X](#) [Windows](#)

STEP 4 Get your Authorization Key

[Generate Authorization Key](#)

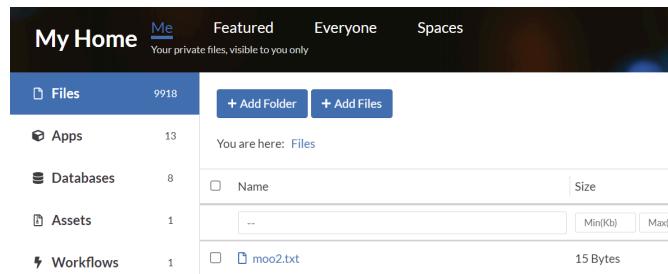


The screenshot shows the 'My Home' dashboard with the 'Files' section selected. A file named 'foo2.txt' is selected, showing its details: LOCATION: Private, ID: file-GJv1zKj0Kj2vzFP4Gg475ZyX-1, ADDED BY: Omar Serang, ORIGIN: Uploaded, FILE SIZE: 15 Bytes, CREATED ON: 2022-11-19 00:13:47 UTC.

```
pfda download -key
Mk5VTEN1TS83R2I1U3dXQkRnWEhzamJvVVFrTVZrOHA4STI4OTM0Mi tRWnNqZWVBSVRnd
lBicG1IUU9PeStjbTBLRXUzNW5rMmMrMjV6bGVhSnVTUlhd2dEOvhRdUZvdmE1a29pcH
dWWS92RGNyN11jT1ZtdnNjbE15RXVyVn11Zkd3UVVxODZpYzNsWi9JWVVBCeW3VE5uaXd
MSTdYNHNWVFJpZGJYdX1Va2hsRFFnR2dDc1JISzhuYWxla2JXLs1zVjRhSVBCWFdaRXBF
WnBsMXNtSXB3PT0=--e19f53de7644d63dd3898717896a88bd0a383db6 -file-id
file-GJv1zKj0Kj2vzFP4Gg475ZyX-1
```

Upload a file from the workstation local filesystem to precisionFDA (note the key is cached).

```
mv foo2.txt moo2.txt
pfda upload-file -file moo2.txt
```



The screenshot shows the 'My Home' dashboard with the 'Files' section selected. A file named 'moo2.txt' is uploaded, showing its details: You are here: Files, Name: moo2.txt, Size: 15 Bytes.

Deploy Local PostgreSQL DB Server

Deploy a local PostgreSQL DB server on the Data Analysis workstation. Map the postgres port from the container to the workstation (host) OS. In the notebook terminal:

```
sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt
$(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc
| apt-key add -
apt-get update
apt-get -y install postgresql < "/dev/null"
```

Configure postgres to enable password-free local login. Find pg_hba.conf in /etc/postgresql/ and configure with permissive permissions.

```
find /etc/postgresql -name pg_hba.conf | xargs sed -i 's/peer/trust/'
find /etc/postgresql -name pg_hba.conf | xargs sed -i 's/md5/trust/'
```

Start the local PostgreSQL DB server on the Data Analysis notebook.

```
/etc/init.d/postgresql start
 * Starting PostgreSQL 15 database server
[ OK ]
/etc/init.d/postgresql status
15/main (port 5432): online

psql -U postgres -h 127.0.0.1
psql (15.1 (Ubuntu 15.1-1.pgdg18.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384,
compression: off)
Type "help" for help.

postgres=#
```

Create a Table with some data in the Local DB

In psql in the notebook terminal create a table, then copy two records from stdin into the table display them.

```
CREATE TABLE public."PATIENT" (
    patient_id int NOT NULL,
    name character varying,
    gender character varying,
    zip character varying,
    country character varying,
    created_date date
);

COPY public."PATIENT" (patient_id, name, gender, zip, country,
created_date) from stdin;
```

Add these two records when prompted from the above COPY, and terminate with the \. record.

```
12345 foo    m      94040 usa    2022-11-25
54321 bar    m      94040 usa    2022-11-25
\.
```

```
select * from public."PATIENT";
```

```

patient_id | name | gender | zip | country | created_date
-----+-----+-----+-----+-----+-----
 12345 | foo  | m    | 94040 | usa   | 2022-11-25
 54321 | bar  | m    | 94040 | usa   | 2022-11-25
(2 rows)

```

Create a Notebook and Connect to the Local DB

In the notebook terminal, install the psycopg2 binary.

```
pip install psycopg2-binary
```

Open a Python 3 notebook.



And enter the following code:

```

import psycopg2
conn = psycopg2.connect("dbname='postgres' user='postgres'
host='127.0.0.1'")
cur = conn.cursor()
cur.execute('SELECT * FROM public."PATIENT" limit 10')
# fetch results
rows = cur.fetchall()
# iterate through results
for row in rows:
    print (" ", row)

```

The screenshot shows a Jupyter Notebook interface with a single code cell labeled [2]. The cell contains the same Python code as above. When run, the output is displayed below the cell, showing the two rows of patient data from the PostgreSQL database.

```

[2]: import psycopg2
      conn = psycopg2.connect("dbname='postgres' user='postgres' host='127.0.0.1'")
      cur = conn.cursor()
      cur.execute('SELECT * FROM public."PATIENT" limit 10')
      # fetch results
      rows = cur.fetchall()
      # iterate through results
      for row in rows:
          print (" ", row)

(12345, 'foo', 'm', '94040', 'usa', datetime.date(2022, 11, 25))
(54321, 'bar', 'm', '94040', 'usa', datetime.date(2022, 11, 25))

```

Connect to the Cluster DB

Open a Python 3 notebook.



And enter the following code:

```

import psycopg2

conn =
psycopg2.connect("dbname='workstations_and_databases_tutorial_db'
user='root'
host='dbcluster-gk0b58j0kj2y4v1k899bq0x6.cluster-cqy4cenhebvb.us-east
-1.rds.amazonaws.com' password='password'")

cur = conn.cursor()
cur.execute('SELECT * FROM public."PATIENT" limit 10')
# fetch results
rows = cur.fetchall()
# iterate through results
for row in rows:
    print ("PATIENT", row[0], row[1], row[2])

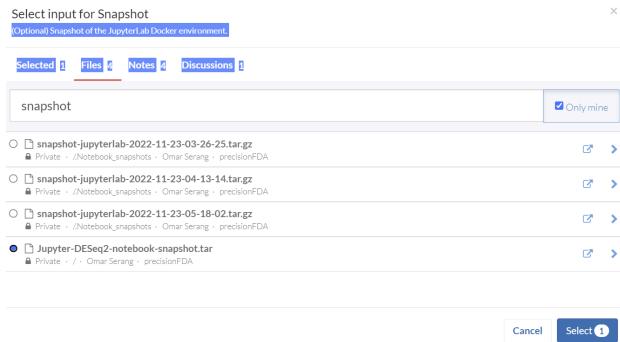
cur.execute('SELECT * FROM public."OBSERVATION" limit 10')
# fetch results
rows = cur.fetchall()
# iterate through results
for row in rows:
    print ("OBSERVATION", row[0], row[1], row[2])

PATIENT 12345 Fred Foobar M
PATIENT 12346 Mary Merry F
PATIENT 12347 Barney Rubble M
OBSERVATION 9870 12345 Annual check up
OBSERVATION 9871 12345 Emergency
OBSERVATION 9872 12346 Clinic visit
OBSERVATION 9873 12347 Lab results
OBSERVATION 9874 12347 Post-op checkup

```

Load a Complete Notebook from a Snapshot

Using My Home / Applications, run the featured pfda-jupyterLab app on the smallest instance type, providing the *Jupyter-DESeq2-notebook-snapshot.tar* file as input. This snapshot contains a complete RNA-seq DESeq2 quantification JupyterLab workbook with R package, notebook, input file and sample sheet all included.



Once the app is running, click the open workstation button to access a rich visual and interactive analysis environment.

The screenshot shows the 'My Home' dashboard with the following details:

- Me**: Your private executions, visible to you only.
- Files**: 9921
- Apps**: 13
- Databases**: 9
- Assets**: 1
- Workflows**: 1
- Executions**: 192

RNA-seq DESeq2 Notebook (running)

Actions: Open Workstation, Sync Files, Re-Run Execution, Actions

LOCATION: APP
Private: DNAAnexus JupyterLab Server on precisionFDA

LAUNCHED BY: Omar Serang

CREATED ON: 2022-11-26 22:35:37 UTC

INSTANCE TYPE: mem1(ssd1_x2_fedramp)

DURATION: N/A **COST**: \$ **APP REVISION**: 2

PROJECTS APPS

File Edit DNAAnexus View Run Kernel Tabs Settings Help

Launcher

Notebook

Python 3

R

Console

Python 3

R

File

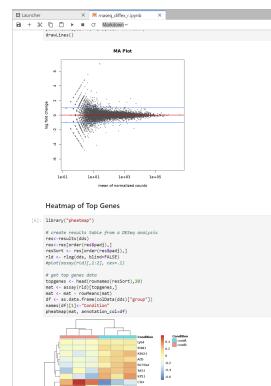
+

DNAAnexus

Name **Last Modified**

- maseq_differ_t.ipynb** 6 days ago
- condA_1.ct 4 years ago
- condA_2.ct 4 years ago
- condA_3.ct 4 years ago
- condB_1.ct 4 years ago
- condB_2.ct 4 years ago
- condB_3.ct 4 years ago
- create_jupyterlab_se...** 4 years ago
- maseq_samples.tsv** 2 years ago

Open the *rnaseq_diffex_r* notebook to explore the data.



Build Epidemiology Data Analysis Notebooks

The Epidemiologist's R Handbook (<https://epirhandbook.com/en>) provides a wealth of R code examples for applied epidemiology and public health. The following pipelines have been created for deployment on Jupyter Notebooks using snapshots:

- Data cleaning and de-duplication
- Time series and outbreak detection
- Epidemic modeling and contact tracking

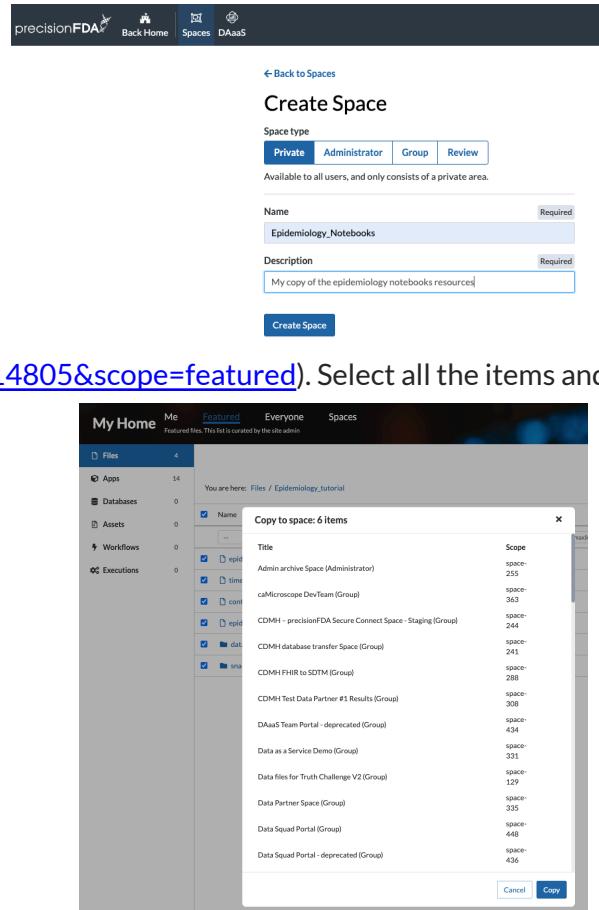
Copy the Notebook Resources to a New Private Space

In order to access these resources from the Notebooks that you launch, you will need to copy them to a private space. Create a new private space for this purpose.

The resources to launch and test these pipelines are available in the Epidemiology_tutorial folder that can be accessed under My Home / Files / Featured

(https://precision.fda.gov/home/files?folder_id=8314805&scope=featured). Select all the items and copy to your new private space using the Actions pulldown menu.

You only have to perform this operation once.



The screenshot shows the precisionFDA interface. At the top, there is a navigation bar with the precisionFDA logo, Back Home, Spaces, and DAaaS. Below the navigation bar, a 'Create Space' dialog is open. It has tabs for Private, Administrator, Group, and Review, with Private selected. The 'Space type' is set to Private. The 'Name' field contains 'Epidemiology_Notebooks'. The 'Description' field contains 'My copy of the epidemiology notebooks resources'. A 'Create Space' button is at the bottom. In the background, there is a 'My Home' dashboard with sections for Apps, Databases, Assets, Workflows, and Executions. A 'Copy to space: 6 items' modal is overlaid on the dashboard. The modal lists six items with their titles and scopes:

Title	Scope
Admin archive Space (Administrator)	space-255
caMicroscope DevTeam (Group)	space-363
CDMH - precisionFDA Secure Connect Space - Staging (Group)	space-244
CDMH-database transfer Space (Group)	space-241
CDMH-FHIR to SDTM (Group)	space-288
CDMH Test Data Partner #1 Results (Group)	space-308
DAaaS Team Portal - deprecated (Group)	space-434
Data as a Service Demo (Group)	space-331
Data files for Truth Challenge V2 (Group)	space-129
Data Partner Space (Group)	space-335
Data Squad Portal (Group)	space-448
Data Squad Portal - deprecated (Group)	space-436

Running in non-interactive mode with papermill

The epidemic_modeling.ipynb notebook can take a considerable amount of time to run (1.5 hours on a Baseline 16 instance) and thus has issues when attempting to run it interactively. It is recommended that you first run the notebook non-interactively using "papermill". This is a two-step process, first running the notebook non-interactively and then opening the rendered result in a new notebook.

You will need the file ID for the epidemic_modeling.ipynb notebook. Remove the -x suffix for use in the runtime command invocation (e.g. file-GYXjgGj0J85qJ8x618yybg8v).

Epidemiology_Notebooks

My copy of the epidemiology notebooks resources

Private Area		
Files	22	
	Apps	0
	Workflows	0
	Executions	4
	Members	1
	Reports	0

Back to Files

epidemic_modeling.ipynb

No description provided.

LOCATION: Epidemiology_Notebooks - Private ID: file-GYXjgGj0J85qJ8x618yybg8v-4

Back to App Run App: DNAAnexus JupyterLab Server on precisionFDA

? Need help? Learn more about running an app

CONFIGURE

Job Name	Required	Execution Cost Limit (\$)	Required
Epidemic Modeling Notebook Batch Mode	100		

Context: Epidemiology_Notebooks (Private) - space-539 Required Instance Type Required Baseline 16.2.288\$/hour Maximum estimated runtime: 43h 42m

INPUTS

Duration	2400
(Optional) Initial duration of the JupyterLab interactive environment in minutes. Ignored when cmd argument is specified.	
Image name	(Optional) Name of a Docker image, available in a Docker registry (e.g. DockerHub, Quay.io).
Snapshot	1 File Selected Clear (Optional) Snapshot of the JupyterLab Docker environment.
Command line	dx download file-GYXjgGj0J85qJ8x618yybg8v && papermill epidemic_modeling.ipynb pfda_epidemic_modeling_eval.ipynb
(Optional) Command to execute in the JupyterLab environment. View the app Readme for details.	

Follow the procedure in the [Run the pfda-jupyterLab Featured App](#) section of this tutorial, selecting the execution Context as Epidemiology_Notebooks, specifying snapshot-epidemic-modeling.tar.gz in the Snapshot inputs section, and specifying the following in the Command Line input:

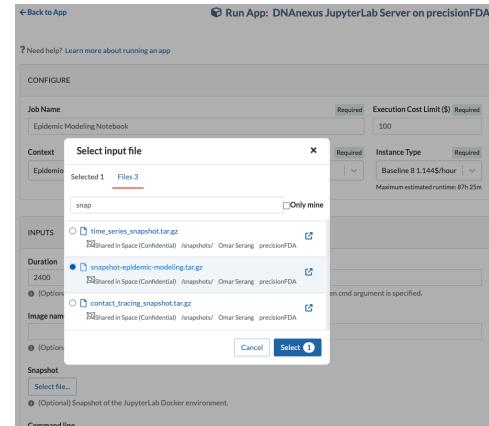
```
dx download file-GYXjgGj0J85qJ8x618yybg8v && papermill
epidemic_modeling.ipynb epidemic_eval.ipynb
```

You'll notice the following error in the job log but be patient, it will complete.

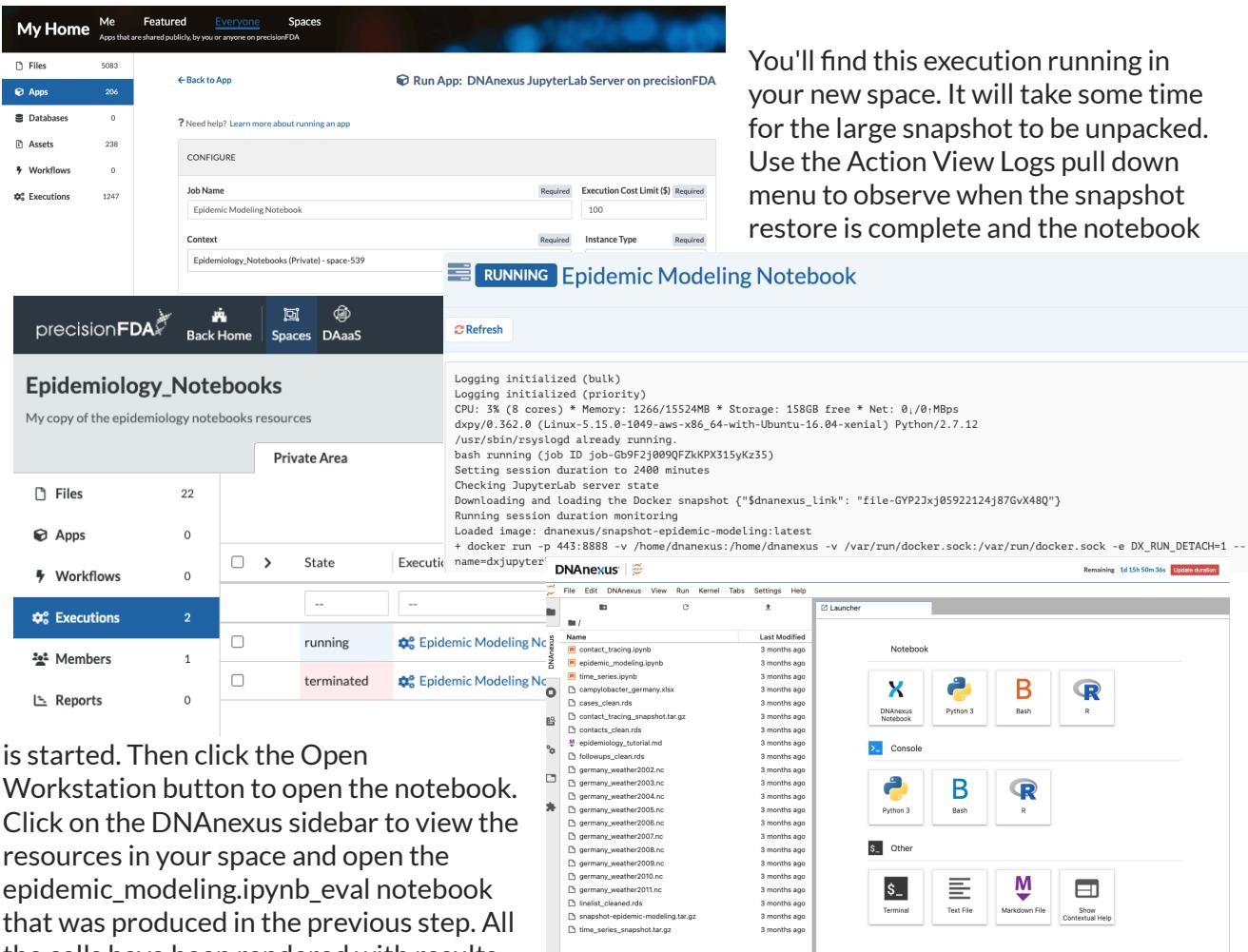
```
Executing: 18% |████| 7/39 [00:01<00:17, 1.79cell/s] Failed to create bus
connection: No such file or directory
```

Build Epidemic Modeling Notebook

Follow the procedure in the [Run the pfda-jupyterLab Featured App](#) section of this tutorial, selecting the execution Context as Epidemiology_Notebooks and specifying snapshot-epidemic-modeling.tar.gz in the Snapshot inputs section, to launch the epidemic modeling notebook.



A screenshot of a 'Select input file' dialog box. The 'Selected 1' dropdown shows 'Files 3'. The 'snapshot-epidemic-modeling.tar.gz' file is highlighted with a blue circle and labeled 'Selected in Space (Confidential) /snapshots/ Omar Serang precisionFDA'. Other files listed are 'time_series_snapshot.tar.gz' and 'contact_tracing_snapshot.tar.gz', both with 'Unselected' status.



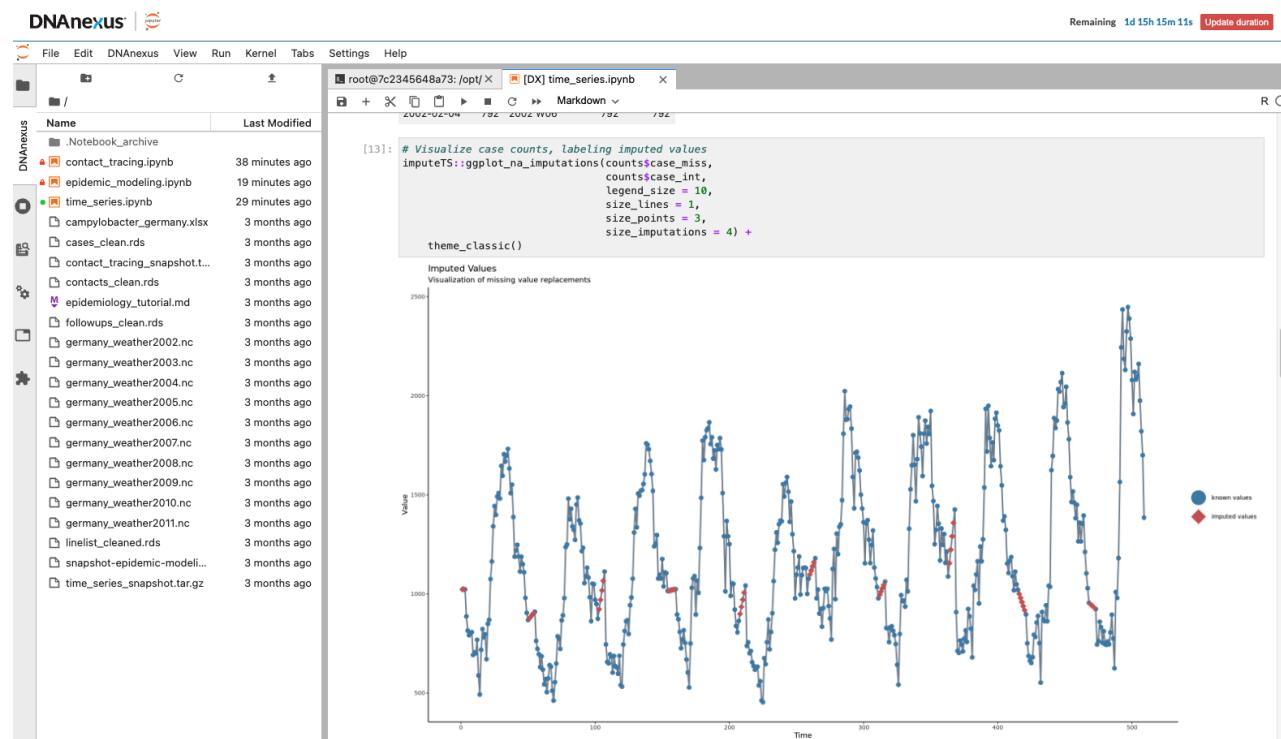
The screenshot shows the precisionFDA interface. On the left, there's a sidebar with 'My Home' selected, showing counts for Files (5083), Apps (206), Databases (0), Assets (238), Workflows (0), and Executions (1247). The main area shows a configuration dialog for running a DNAexus JupyterLab Server. It has fields for 'Job Name' (Epidemic Modeling Notebook), 'Execution Cost Limit (\$)' (100), and 'Context' (Epidemiology_Notebooks (Private) - space-539). Below this, it says 'RUNNING Epidemic Modeling Notebook'. The bottom part shows a list of files in a 'Private Area' under 'Executions', with one entry for 'Epidemic Modeling Notebook' in state 'running'. To the right, a terminal window shows the command used to run the notebook, and a 'Launcher' panel lists various tools like DNAexus Notebook, Python 3, Bash, R, Terminal, Text File, and Markdown File.

You'll find this execution running in your new space. It will take some time for the large snapshot to be unpacked. Use the Action View Logs pull down menu to observe when the snapshot restore is complete and the notebook

is started. Then click the Open Workstation button to open the notebook. Click on the DNAexus sidebar to view the resources in your space and open the `epidemic_modeling.ipynb_eval` notebook that was produced in the previous step. All the cells have been rendered with results presented.

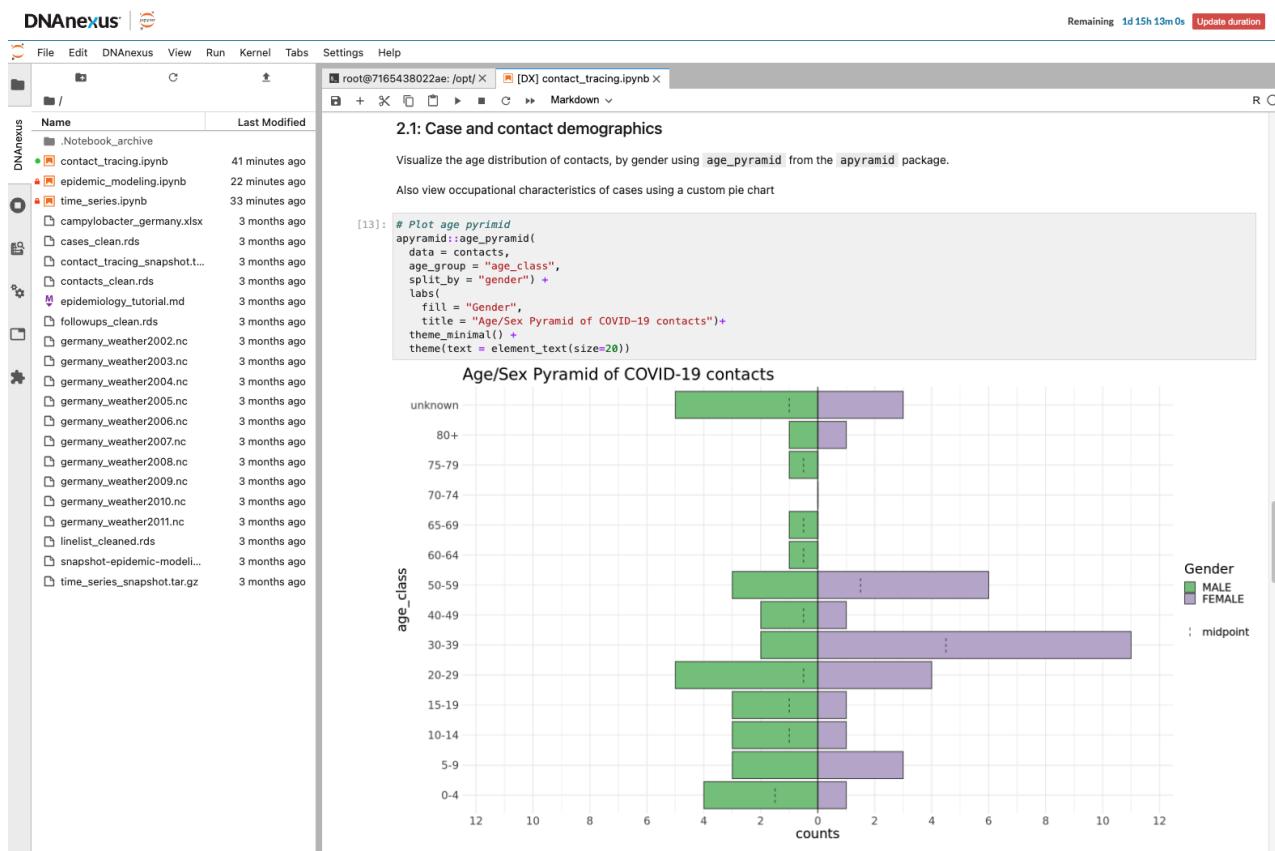
Build Time Series and Outbreak Detection Notebook

Follow the procedure in the [Build Epidemic Modeling Notebook](#) section above, selecting the execution Context as Epidemiology_Notebooks and specifying time_series_snapshot.tar.gz in the Snapshot inputs section, to launch the time series and outbreak detection notebook. Once the notebook is running and the snapshot restored, click on the DNAexus sidebar to view the resources in your space and open the time_series.ipynb notebook. Select Run All Cells from the Run menu and view the calculation results appear in the notebook cells.



Build Contact Tracing Notebook

Follow the procedure in the [Build Epidemic Modeling Notebook](#) section above, selecting the execution Context as Epidemiology_Notebooks and specifying contact_tracing_snapshot.tar.gz in the Snapshot inputs section, to launch the contract tracing notebook. Once the notebook is running and the snapshot restored, click on the DNAexus sidebar to view the resources in your space and open the contact_tracing.ipynb notebook. Select Run All Cells from the Run menu and view the calculation results appear in the notebook cells.



Snapshot and Terminate Workstations

In keeping with good cloud usage practice, we will snapshot and terminate the workstations, preserving their entire state as built out through this tutorial. Additionally since we've backed up the database to a precisionFDA file, we can safely terminate the cluster database as well.

Stop the Docker Containers and Snapshot Data Analysis Workstation

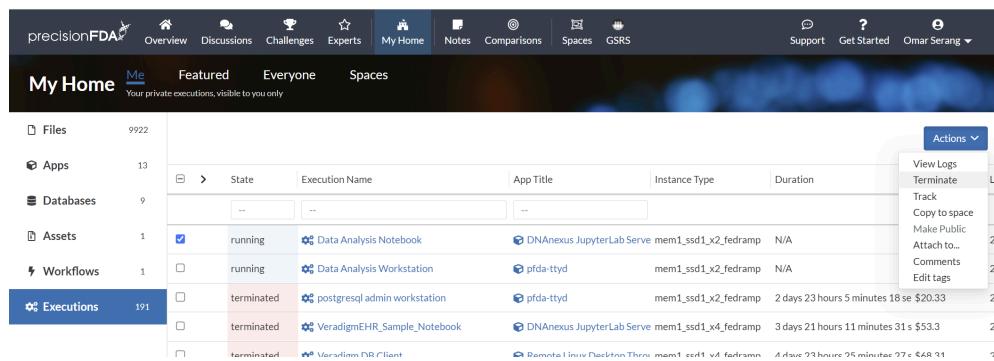
Using the data analysis workstation shell, create a snapshot of the workstation in you My Home files area.

```
Docker stop
dx-create-snapshot

dx ls -al *snapshot
```

Terminate the Workstation

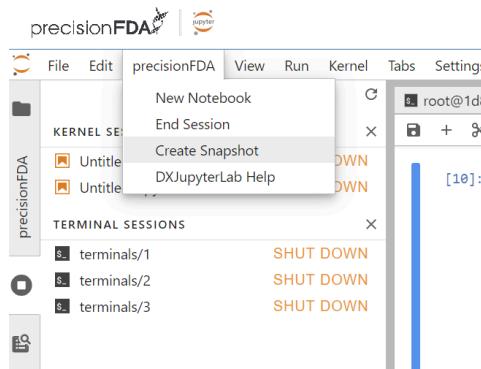
In My Home / Executions, select (one at a time unfortunately) the Data Analysis Workstation and Data Analysis Notebook executions and select Terminate under the Action dropdown menu.



	State	Execution Name	App Title	Instance Type	Duration
running	Data Analysis Notebook	DNAexus JupyterLab Serve	mem1_ssdl1_x2_fedramp	N/A	
running	Data Analysis Workstation	pfda-ityd	mem1_ssdl1_x2_fedramp	N/A	
terminated	postgresql admin workstation	pfda-ityd	mem1_ssdl1_x2_fedramp	2 days 23 hours 5 minutes 18 seconds \$20.33	
terminated	VeradigmEHR_Sample_Notebook	DNAexus JupyterLab Serve	mem1_ssdl1_x4_fedramp	3 days 21 hours 11 minutes 31 seconds \$53.3	
terminated	Varsilium DR Client	Docker Linux Container	mem1_e811_v4_fedramp	4 days 22 hours 56 minutes 27 seconds \$48.31	

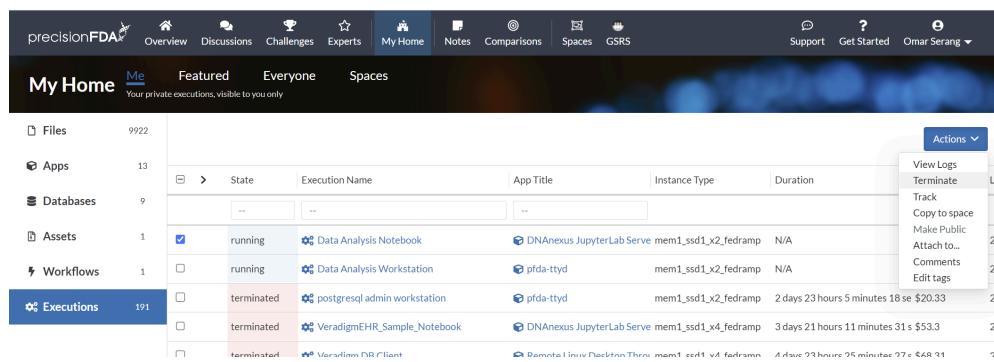
Snapshot and Terminate the Data Analysis Notebook

Select Create Snapshot in the precisionFDA menu in the jupyterLabs interface.



Terminate the Workstation and Notebook and Database Cluster

In My Home / Executions, select (one at a time unfortunately) the Data Analysis Workstation and Data Analysis Notebook executions and select Terminate under the Action dropdown menu.



The screenshot shows the 'My Home' dashboard with the 'Me' tab selected. The 'Executions' section is highlighted, showing 191 entries. A context menu is open over the first execution in the list, with the 'Actions' dropdown expanded. The menu items are:

- View Logs
- Terminate**
- Track
- Copy to space
- Make Public
- Attach to..
- Comments
- Edit tags

	State	Execution Name	App Title	Instance Type	Duration
<input checked="" type="checkbox"/>	running	Data Analysis Notebook	DNAexus JupyterLab Serve	mem1_ssdl1_x2_fedramp	N/A
<input type="checkbox"/>	running	Data Analysis Workstation	pfda-ttyd	mem1_ssdl1_x2_fedramp	N/A
<input type="checkbox"/>	terminated	postgresql admin workstation	pfda-ttyd	mem1_ssdl1_x2_fedramp	2 days 23 hours 5 minutes 18 se \$20.33
<input type="checkbox"/>	terminated	VeradigmEHR_Sample_Notebook	DNAexus JupyterLab Serve	mem1_ssdl1_x4_fedramp	3 days 21 hours 11 minutes 31 s \$53.3
<input type="checkbox"/>	terminated	Versitum DR Client	Remote Linux Desktop Through	mem1_ecl1_v1_fedramp	4 days 23 hours 25 minutes 77 s \$48.31