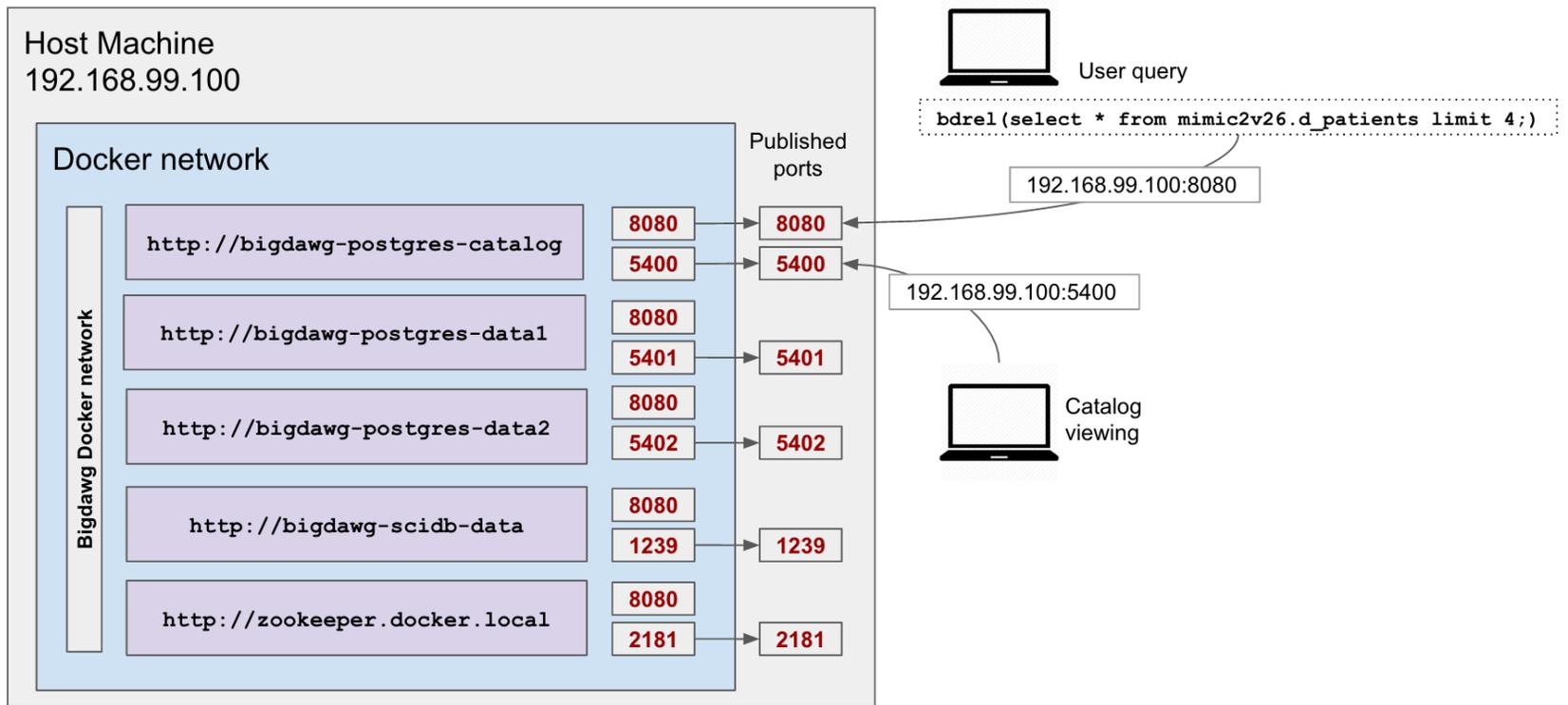


# Hands-On Overview:

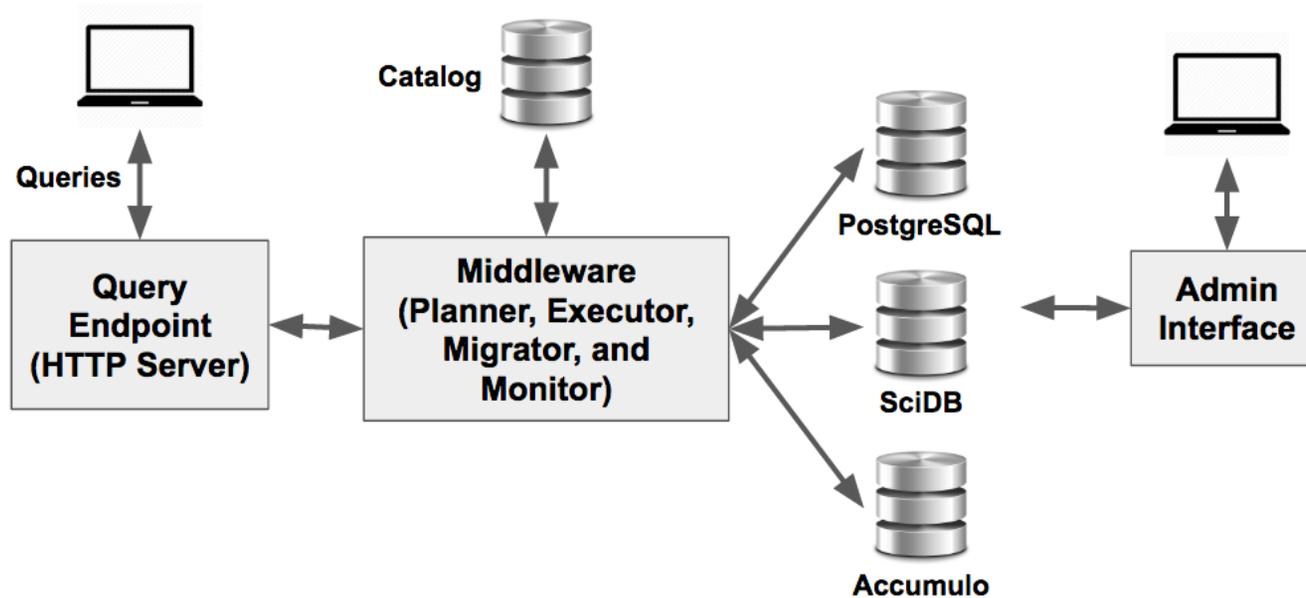
- Demo cluster architecture
- Working with the demo cluster: Docker basics
  - List the running containers
  - Logging into a container
- Explore the Catalog and view the Postgres tables
- BigDAWG Use cases:
  - Postgres to Postgres with BigDAWG
  - SciDB to Postgres with BigDAWG
  - Postgres to Accumulo with BigDAWG

## Demo Cluster Architecture

## Docker Container Networking Diagram



## Docker Container Descriptions



#### **bigdawg-postgres-catalog**

Runs the Catalog, Middleware, and Query Endpoint. The Query Endpoint listens for queries on `bigdawg-postgres-catalog` and port `8080`

#### **bigdawg-postgres-data1**

Runs PostgreSQL loaded with the MIMIC II patient dataset

#### **bigdawg-postgres-data2**

Runs PostgreSQL loaded with a copy of the Mimic II patient dataset. Used for demonstrating migration between 2 PostgreSQL instances

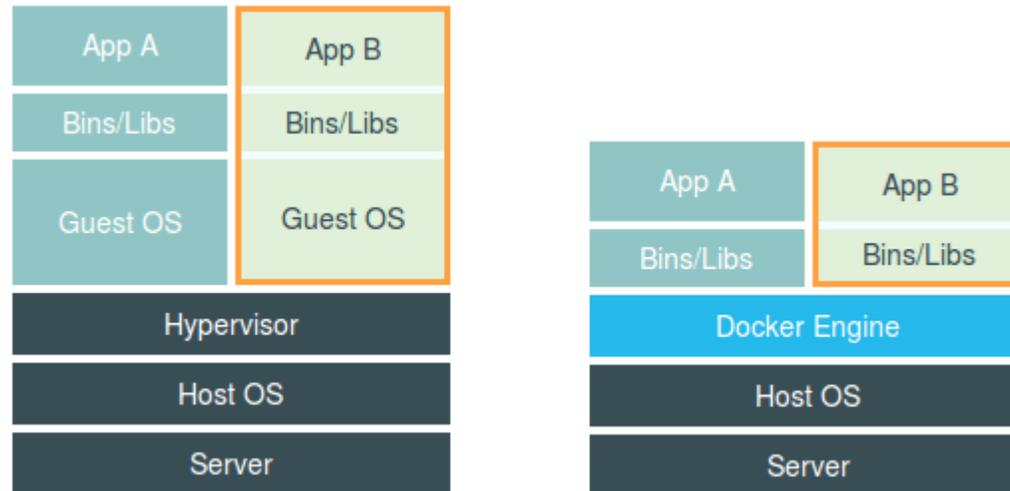
#### **bigdawg-sciadb**

Runs SciDB with MIMIC II waveform data

#### **Accumulo containers: several containers support the Accumulo stack:**

bigdawg-accumulo-master: Master server  
 bigdawg-accumulo-tserver0: Handles client reads and writes  
 bigdawg-accumulo-zookeeper  
 bigdawg-accumulo-namenode  
 bigdawg-accumulo-proxy

# Working with the demo cluster: Docker basics



Virtual Machines:

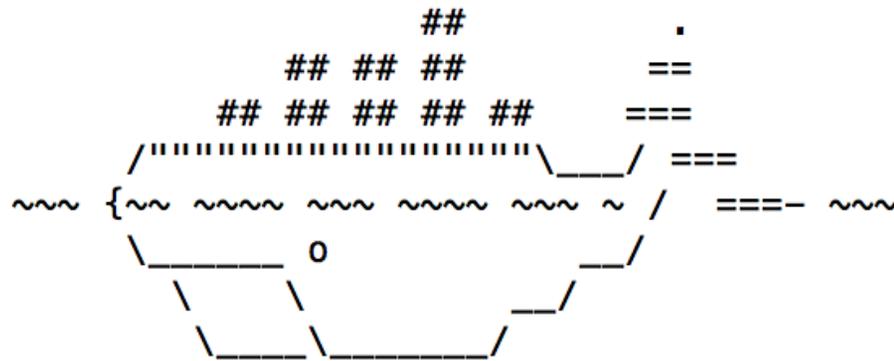
- Each application includes separate OS and resources that are not shared.

Docker:

- Each application is run as an isolated processes on the host OS and shares its resources.

## Open a Docker Quickstart Terminal

Open a new Docker Quickstart Terminal. This is required to run any docker commands.



`docker` is configured to use the `default` machine with IP `192.168.99.100`  
For help getting started, check out the docs at <https://docs.docker.com>

## List the containers running on Docker

A list of the running containers is shown. Containers can be referred to by container ID or name. The ports show which port local to a container is forwarded to the docker host port.

## Docker command:

```
> docker ps
```

## Output:

```
docker ps
CONTAINER ID          IMAGE                COMMAND              CREATED
9a041c769cfb         bigdawg/accumulo    "/usr/bin/supervisord" 2 hours ago
59e91c8a95f1         bigdawg/accumulo    "/usr/bin/supervisord" 2 hours ago
1aa16f0d88a1         bigdawg/accumulo    "/usr/bin/supervisord" 2 hours ago
52f26c00d069         bigdawg/accumulo    "/usr/bin/supervisord" 2 hours ago
b782323ba115         bigdawg/accumulo    "/usr/bin/supervisord" 2 hours ago
5f3a08ff00c4         bigdawg/scidb       "/bin/sh -c /start_se" 2 hours ago
a44b212568da         bigdawg/postgres    "/bin/sh -c /start_se" 2 hours ago
c5e6b9d51bc9         bigdawg/postgres    "/bin/sh -c /start_se" 2 hours ago
94d7cdcdeace         bigdawg/postgres    "/bin/sh -c /start_se" 2 hours ago
```

(continued)

```
STATUS          PORTS                NAMES
Up 2 hours     0.0.0.0:42424->42424/tcp  bigdawg-accumulo-proxy
Up 2 hours     0.0.0.0:9999->9999/tcp,
               0.0.0.0:50095->50095/tcp  bigdawg-accumulo-master
Up 2 hours     0.0.0.0:9997->9997/tcp  bigdawg-accumulo-tserver0
Up 2 hours     0.0.0.0:2181->2181/tcp  bigdawg-accumulo-zookeeper
Up 2 hours     0.0.0.0:1239->1239/tcp  bigdawg-scidb-data
Up 2 hours     0.0.0.0:5402->5402/tcp  bigdawg-postgres-data2
Up 2 hours     0.0.0.0:5401->5401/tcp  bigdawg-postgres-data1
Up 2 hours     0.0.0.0:5400->5400/tcp,
               0.0.0.0:8080->8080/tcp  bigdawg-postgres-catalog
```

# Log into a docker container:

Docker command:

```
> docker exec -it bigdawg-postgres-catalog bash
```

Output:

```
postgres@bigdawg-postgres-catalog:
```

## Explore the Catalog and view the Postgres tables

After logging into the `bigdawg-postgres-catalog` container, use `psql` to navigate the catalog database view the tables.

PSQL Command:

```
> psql bigdawg_catalog
```

Output:

```
psql (9.4.10)  
Type "help" for help.
```

```
bigdawg_catalog=#
```

You are logged in as user `postgres` with password `postgres`.

## Verify connection info:

## PSQL Command:

```
> \conninfo
```

```
bigdawg_catalog=# \conninfo
```

```
You are connected to database "bigdawg_catalog" as user "postgres" via socket in  
"/var/run/postgresql" at port "5400".
```

## Show list of relations in catalog schema:

The Catalog stores metadata about the database engines and data objects under its control. This metadata is stored in the following tables.

## PSQL Command:

```
> \dt catalog.
```

## Output:

```
bigdawg_catalog=# \dt catalog.
```

```
          List of relations
```

Schema	Name	Type	Owner
catalog	casts	table	postgres
catalog	databases	table	postgres
catalog	engines	table	postgres
catalog	islands	table	postgres
catalog	objects	table	postgres
catalog	scidbbinpaths	table	postgres
catalog	shims	table	postgres

```
(7 rows)
```

# View rows of data from a table:

SQL Command:

```
> SELECT * FROM catalog.engines;
```

Output:

## catalog.engines

```
bigdawg_catalog=# SELECT * FROM catalog.engines;
```

eid	name	host	port	connection_properties
0	postgres0	bigdawg-postgres-catalog	5400	PostgreSQL 9.4.5
1	postgres1	bigdawg-postgres-data1	5401	PostgreSQL 9.4.5
2	postgres2	bigdawg-postgres-data2	5402	PostgreSQL 9.4.5
3	scidb_local	bigdawg-scidb-data	1239	SciDB 14.12
4	saw ZooKeeper	zookeeper.docker.local	2181	Accumulo 1.6

(5 rows)

## catalog.databases

```
bigdawg_catalog=# SELECT * FROM catalog.databases;
 dbid | engine_id |      name      |  userid  | password
-----+-----+-----+-----+-----
    0 |          0 | bigdawg_catalog | postgres | test
    1 |          0 | bigdawg_schemas | postgres | test
    2 |          1 | mimic2          | postgres | test
    3 |          2 | mimic2_copy     | postgres | test
    4 |          0 | tpch            | postgres | test
    5 |          1 | tpch            | postgres | test
    6 |          3 | scidb_local     | scidb    | scidb123
    7 |          4 | accumululo      | bigdawg  | bigdawg
(8 rows)
```

### **catalog.islands**

```
bigdawg_catalog=# SELECT * FROM catalog.islands;
 iid | scope_name | access_method
-----+-----+-----
    0 | RELATIONAL | PSQL
    1 | ARRAY      | AFL
    2 | TEXT       | JSON
(3 rows)
```

## **Postgres to Postgres Query Example**

Problem: Data is entered into tables on two separate Postgres engines, and we need to perform a join across those tables.

1. `admissions`: Times when patients were admitted

## 2. additives: Medication types, dosage, and administration times

We need to know when patients were admitted from `admissions` and when their medication was delivered from `additives` so we can see how long they waited until the first medication was administered.

### Table 1: admissions table

Contains the admit and discharge times for each patient.

BigDAWG Syntax:

```
bdrel(SELECT subject_id, admit_dt, disch_dt
      FROM mimic2v26.admissions
      LIMIT 5);
```

Output:

```
> curl -X POST -d "bdrel(SELECT subject_id, admit_dt, disch_dt FROM mimic2v26.admissions LIMIT 5)" http://192.168.99.100:8080/bigdawg/query/
```

subject_id	admit_dt	disch_dt
56	2644-01-17 00:00:00.0	2644-01-23 00:00:00.0
78	2778-03-24 00:00:00.0	2778-03-27 00:00:00.0
3	2682-09-07 00:00:00.0	2682-09-18 00:00:00.0
12	2875-09-26 00:00:00.0	2875-10-09 00:00:00.0
26	3079-03-03 00:00:00.0	3079-03-10 00:00:00.0

## Table 2: additives table

Contains the medications delivered to each patient.

BigDAWG Syntax:

```
bdrel(SELECT subject_id, charttime, amount, route
      FROM mimic2v26.additives
      LIMIT 10;)
```

Output:

```
> curl -X POST -d "bdrel(SELECT subject_id, charttime, amount, route FROM mimic2v
26.additives LIMIT 10;)" http://192.168.99.100:8080/bigdawg/query/
```

subject_id	charttime	amount	route
78	2778-03-24 04:00:00.0	1.0	IV Drip
31	2678-08-22 13:00:00.0	40.0	Intravenous Push
31	2678-08-22 13:00:00.0	60.0	IV Drip
31	2678-08-22 15:00:00.0	1250.0	IV Drip
31	2678-08-23 15:00:00.0	20.0	Intravenous Push

## Location Transparency

The Catalog knows on which engine the tables reside:

- admissions table is on logical\_db==2 and physical\_db==2
- additives table is on logical\_db==2 and physical\_db==3

Entry from catalog.objects:

oid	name	fields	logical_db	physical_db
3	"mimic2v26.additives"	"subject_id,icustay_id,..."	2	3
4	"mimic2v26.admissions"	"hadm_id,subject_id,..."	2	2

## Postgres to Postgres BigDAWG Query

Solution: Query the Relational island for data stored in different tables.

BigDAWG Syntax:

```
bdrel(SELECT mimic2v26.additives.subject_id,  
        mimic2v26.admissions.admit_dt,  
        mimic2v26.additives.charttime,  
        mimic2v26.additives.route  
        FROM mimic2v26.additives, mimic2v26.admissions  
        WHERE mimic2v26.additives.subject_id=mimic2v26.admissions.subject_id  
        LIMIT 15;)
```

Output:

```
> curl -X POST -d "bdrel(SELECT mimic2v26.additives.subject_id, mimic2v26.admissions.admit_dt, mimic2v26.additives.charttime, mimic2v26.additives.route FROM mimic2v26.additives,mimic2v26.admissions WHERE mimic2v26.additives.subject_id=mimic2v26.admissions.subject_id LIMIT 15;)" http://192.168.99.100:8080/bigdawg/query/
```

subject_id	admit_dt	charttime	route
3	2682-09-07 00:00:00.0	2682-09-08 03:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-08 03:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-08 03:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-08 17:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-08 16:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-08 13:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-08 06:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-07 22:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-07 22:00:00.0	IV Drip
3	2682-09-07 00:00:00.0	2682-09-07 22:00:00.0	IV Drip
12	2875-09-26 00:00:00.0	2875-09-28 01:00:00.0	Intravenous Push
12	2875-09-26 00:00:00.0	2875-09-27 04:00:00.0	IV Drip
12	2875-09-26 00:00:00.0	2875-09-27 09:00:00.0	IV Drip
21	3139-03-19 00:00:00.0	3138-10-30 16:00:00.0	Intravenous Push
21	3138-10-29 00:00:00.0	3138-10-30 16:00:00.0	Intravenous Push

## SciDB and Postgres Example

Problem: Waveworm data exists in SciDB and we want to move peak values to Postgres where we can join it later with other records.

1. SciDB: myarray contains timeseries of electrocardiogram (ECG) data.

# View SciDB Data

Log into scidb container and start iquery:

```
> docker exec -it bigdawg-scidb-data bash
> su scidb
> cd /home/scidb
> iquery
```

AQL%

List all arrays:

```
> AQL% select * from list('arrays');

{No} name,uid,aid,schema,availability,temporary
{0} 'myarray',1,1,'myarray<dim1:double,dim2:double> [i=0:*,1000000,0]',true,false
{1} 'test_array',4,4,'test_array<val:int64> [i=0:*,100000,0,j=0:*,100000,0]',true,false
{2} 'test_array_flat',3,3,'test_array_flat<val:int64,i:int64,j:int64> [i_0:*,100000,0]',true,false
```

Select myarray:



Output:

```
> curl -X POST -d "bdrel(SELECT * FROM bdcast( bddarray(filter(myarray,dim1>150)),
  res, '(i bigint, dim1 real, dim2 real)', relational))" http://192.168.99.100:8080/bigdawg/query/
```

i	dim1	dim2
37	151.273	0.0
47	151.273	0.0
115	174.545	0.0
159	151.273	0.0
165	174.545	0.0
183	174.545	0.0
191	151.273	0.0
193	174.545	0.0
201	174.545	0.0
223	151.273	0.0
225	174.545	0.0
283	151.273	0.0
...		

## Accumulo and Postgres Example

Problem: Clinical reports are in Postgres and we need to persist some of it into Accumulo to organize it with the rest of the text data.

## View the Postgres Data

## BigDAWG Query Syntax:

```
bdrel(  
  SELECT text  
  FROM mimic2v26.noteevents  
  LIMIT 1  
)
```

Output:

```
> curl -X POST -d "bdrel(SELECT text FROM mimic2v26.noteevents LIMIT 1)" http://192.168.99.100:8080/bigdawg/query/
```

DATE: [\*\*2976-11-28\*\*] 9:17 PM  
CHEST (PORTABLE AP) Clip # [\*\*Cl  
ip Number (Radiology) 7295\*\*]  
Reason: SOB, decreased BS on R

---

—  
UNDERLYING MEDICAL CONDITION:  
73 year old man with

REASON FOR THIS EXAMINATION:  
SOB, decreased BS on R

---

—  
FINAL REPORT

HISTORY: 73 year old man with shortness of breath and decreased breath sounds on the right.

COMPARISON: [\*\*2974-3-8\*\*].

PORTABLE CHEST RADIOGRAPH: Again seen is cardiomegaly unchanged. The aorta is tortuous, unchanged. The pulmonary vascularity is normal in appearance without failure. The lungs are clear without focal consolidations or effusions. There is some minimal linear atelectasis at the left base.

IMPRESSION: No congestive heart failure or focal consolidations.

# Postgres to Accumulo Query

Query Postgres data for text and cast to accumulo.

BigDAWG Query Syntax:

```
bdtext(                                     # bdtext(): text island scope
  {'op': 'scan',                             # accumulo syntax
   'table': 'bdcast(                          # bdcast() within the accumulo query
     bdrel(                                    # bdrel(): relational island scope
       SELECT text FROM mimic2v26.noteevents LIMIT 1
     ),
     res,                                     # intermediate CAST object name
     '',                                     # destination island schema
     text)'                                  # CAST destination type
  }
)
```

Output:

```
> curl -X POST -d "bdtext({ 'op' : 'scan', 'table' : 'bdcast(bdrel(select text fr  
om mimic2v26.noteevents limit 1), res, '', text)'})" http://192.168.99.100:8080/b  
igdawg/query/
```

DATE: [\*\*2976-11-28\*\*] 9:17 PM

CHEST (PORTABLE AP)

Clip # [\*\*Clip Number (Radiology) 7295\*

\*]

Reason: SOB, decreased BS on R

---

UNDERLYING MEDICAL CONDITION:

73 year old man with

REASON FOR THIS EXAMINATION:

SOB, decreased BS on R

---

FINAL REPORT

HISTORY: 73 year old man with shortness of breath and decreased breath sounds on the right.

COMPARISON: [\*\*2974-3-8\*\*].

PORTABLE CHEST RADIOGRAPH: Again seen is cardiomegaly unchanged. The aorta is tortuous, unchanged. The pulmonary vascularity is normal in appearance without failure. The lungs are clear without focal consolidations or effusions. There is some minimal linear atelectasis at the left base.

IMPRESSION: No congestive heart failure or focal consolidations.

# Administrative Interface

The Administrative Interface is a Python Flask webapp that interacts with Docker and the Postgres database running the Catalog.

On the "Cluster Status" tab you can start and stop components of the Demo Cluster:

127.0.0.1:5000

BigDAWG Admin Cluster Status Data Catalog Important Links

## Cluster Status

Name	Status		
bigdawg-accumulo-proxy	exited	Start	Stop
bigdawg-accumulo-master	exited	Start	Stop
bigdawg-accumulo-tserver0	exited	Start	Stop
bigdawg-accumulo-zookeeper	exited	Start	Stop
bigdawg-accumulo-namenode	exited	Start	Stop
bigdawg-scidb-data	exited	Start	Stop
bigdawg-postgres-data2	exited	Start	Stop
bigdawg-postgres-data1	exited	Start	Stop
bigdawg-postgres-catalog	exited	Start	Stop

# Administrative Interface

The Administrative Interface is a Python Flask webapp that interacts with Docker and the Postgres database running the Catalog.

On the "Data Catalog" tab you can view the Catalog tables:

## Engines

Engine ID	Name	Host	Port	Connection Properties
0	postgres0	bigdawg-postgres-catalog	5400	PostgreSQL 9.4.5
1	postgres1	bigdawg-postgres-data1	5401	PostgreSQL 9.4.5
2	postgres2	bigdawg-postgres-data2	5402	PostgreSQL 9.4.5
3	scidb_local	bigdawg-scidb-data	1239	SciDB 14.12
4	saw ZooKeeper	zookeeper.docker.local	2181	Accumulo 1.6

## Data Objects

Object ID	Table Name	Contents / Schema
0	mimic2v26.a_chartdurations	subject_id,icustay_id,itemid,elemid,starttime,startrealtime,endtime,cuid,duration
1	mimic2v26.a_iodurations	subject_id,icustay_id,itemid,elemid,starttime,startrealtime,endtime,cuid,duration
2	mimic2v26.a_medddurations	subject_id,icustay_id,itemid,elemid,starttime,startrealtime,endtime,cuid,duration
3	mimic2v26.additives	subject_id,icustay_id,itemid,ioitemid,charttime,elemid,cgid,cuid,amount,doseunits,route
4	mimic2v26.admissions	hadm_id,subject_id,admit_dt,disch_dt

# Shutdown the Demo Cluster

When finished, stop and remove the containers with this script:

```
> ./cleanup_containers.sh
```

Additionally, if you're using Docker Toolbox, you can stop the VM running Docker with the following command:

```
> docker-machine stop default
```