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 8889.

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-scidb
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

<table>
<thead>
<tr>
<th>CONTAINER ID</th>
<th>IMAGE</th>
<th>COMMAND</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>9a041c769cfd</td>
<td>bigdawg/accumulo</td>
<td>&quot;/usr/bin/ supervisord&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>59e91c8a95f1</td>
<td>bigdawg/accumulo</td>
<td>&quot;/usr/bin/ supervisord&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>1aa16f0d88a1</td>
<td>bigdawg/accumulo</td>
<td>&quot;/usr/bin/ supervisord&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>52f26c00d069</td>
<td>bigdawg/accumulo</td>
<td>&quot;/usr/bin/ supervisord&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>b782323ba115</td>
<td>bigdawg/accumulo</td>
<td>&quot;/usr/bin/ supervisord&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>5f3a08ff00c4</td>
<td>bigdawg/scidb</td>
<td>&quot;/bin/sh -c /start_se&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>a44b212568da</td>
<td>bigdawg/postgres</td>
<td>&quot;/bin/sh -c /start_se&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>c5e6b9d51bc9</td>
<td>bigdawg/postgres</td>
<td>&quot;/bin/sh -c /start_se&quot;</td>
<td>2 hours ago</td>
</tr>
<tr>
<td>94d7cdcdeace</td>
<td>bigdawg/postgres</td>
<td>&quot;/bin/sh -c /start_se&quot;</td>
<td>2 hours ago</td>
</tr>
</tbody>
</table>

(continued)

<table>
<thead>
<tr>
<th>STATUS</th>
<th>PORTS</th>
<th>NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:42424-&gt;42424/tcp</td>
<td>bigdawg-accumulo-proxy</td>
</tr>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:9999-&gt;9999/tcp, 0.0.0.0:50095-&gt;50095/tcp</td>
<td>bigdawg-accumulo-master</td>
</tr>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:9997-&gt;9997/tcp</td>
<td>bigdawg-accumulo-tserver0</td>
</tr>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:2181-&gt;2181/tcp</td>
<td>bigdawg-accumulo-zookeeper</td>
</tr>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:1239-&gt;1239/tcp</td>
<td>bigdawg-scidb-data</td>
</tr>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:5402-&gt;5402/tcp</td>
<td>bigdawg-postgres-data2</td>
</tr>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:5401-&gt;5401/tcp</td>
<td>bigdawg-postgres-data1</td>
</tr>
<tr>
<td>Up 2 hours</td>
<td>0.0.0.0:5400-&gt;5400/tcp, 0.0.0.0:8080-&gt;8080/tcp</td>
<td>bigdawg-postgres-catalog</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Schema</th>
<th>Name</th>
<th>Type</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>catalog</td>
<td>casts</td>
<td>table</td>
<td>postgres</td>
</tr>
<tr>
<td>catalog</td>
<td>databases</td>
<td>table</td>
<td>postgres</td>
</tr>
<tr>
<td>catalog</td>
<td>engines</td>
<td>table</td>
<td>postgres</td>
</tr>
<tr>
<td>catalog</td>
<td>islands</td>
<td>table</td>
<td>postgres</td>
</tr>
<tr>
<td>catalog</td>
<td>objects</td>
<td>table</td>
<td>postgres</td>
</tr>
<tr>
<td>catalog</td>
<td>scidbbinpaths</td>
<td>table</td>
<td>postgres</td>
</tr>
<tr>
<td>catalog</td>
<td>shims</td>
<td>table</td>
<td>postgres</td>
</tr>
</tbody>
</table>
(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
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:**

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

**Output:**

```
```

<table>
<thead>
<tr>
<th>subject_id</th>
<th>admit_dt</th>
<th>disch_dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>2644-01-17 00:00:00.0</td>
<td>2644-01-23 00:00:00.0</td>
</tr>
<tr>
<td>78</td>
<td>2778-03-24 00:00:00.0</td>
<td>2778-03-27 00:00:00.0</td>
</tr>
<tr>
<td>3</td>
<td>2682-09-07 00:00:00.0</td>
<td>2682-09-18 00:00:00.0</td>
</tr>
<tr>
<td>12</td>
<td>2875-09-26 00:00:00.0</td>
<td>2875-10-09 00:00:00.0</td>
</tr>
<tr>
<td>26</td>
<td>3079-03-03 00:00:00.0</td>
<td>3079-03-10 00:00:00.0</td>
</tr>
</tbody>
</table>
Table 2: additives table

Contains the medications delivered to each patient.

BigDAWG Syntax:

\[ \text{bdrel} \{ \text{SELECT} \ subject\_id, \ charttime, \ amount, \ route \\
FROM \ mimic2v26.additives \\
\text{LIMIT} \ 10; \} \]

Output:

> curl -X POST -d "\text{bdrel} (\text{SELECT} \ subject\_id, \ charttime, \ amount, \ route \ FROM \ mimic2v26.additives \ \text{LIMIT} \ 10; )" \ http://192.168.99.100:8080/bigdawg/query/

<table>
<thead>
<tr>
<th>subject_id</th>
<th>charttime</th>
<th>amount</th>
<th>route</th>
</tr>
</thead>
<tbody>
<tr>
<td>78</td>
<td>2778-03-24 04:00:00.0</td>
<td>1.0</td>
<td>IV Drip</td>
</tr>
<tr>
<td>31</td>
<td>2678-08-22 13:00:00.0</td>
<td>40.0</td>
<td>Intravenous Push</td>
</tr>
<tr>
<td>31</td>
<td>2678-08-22 13:00:00.0</td>
<td>60.0</td>
<td>IV Drip</td>
</tr>
<tr>
<td>31</td>
<td>2678-08-22 15:00:00.0</td>
<td>1250.0</td>
<td>IV Drip</td>
</tr>
<tr>
<td>31</td>
<td>2678-08-23 15:00:00.0</td>
<td>20.0</td>
<td>Intravenous Push</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>oid</th>
<th>name</th>
<th>fields</th>
<th>logical_db</th>
<th>physical_db</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><code>mimic2v26.additives</code></td>
<td>&quot;subject_id,icustay_id,...&quot;</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td><code>mimic2v26.admissions</code></td>
<td>&quot;hadm_id,subject_id,...&quot;</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**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:

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

**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,uaid,aid,schema,availability,temporary
{0} 'myarray',1,1,'myarray<dim1:double,dim2:double> [i=0:*],true,false
{1} 'test_array',4,4,'test_array<val:int64> [i=0:*],1000000,0,j=0:*],true,false
{2} 'test_array_flat',3,3,'test_array_flat<val:int64,i:int64,j:int64> [i_0=0:*],100
000,0],true,false

Select myarray:
> AQL% select * from myarray;

{1} dim1,dim2
{0} 14.1818,0
{1} -81.4545,0
{2} 14.1818,0
{3} -128,0
{4} 14,0
{5} -151.273,0
{6} 13.8182,0
{7} -151.273,0
{8} 13.9091,0
{9} -128,0
{10} 14,0
...

**SciDB to Postgres Query**

BigDAWG Query Syntax:

```plaintext
bdrel(SELECT * FROM
   bdcast(
     bdarray(filter(myarray,dim1>150)),
     res,
     '(i bigint, dim1 real, dim2 real)',
     relational
   )
)
```

# bdrel(): relational island query
# bdcast(): cast syntax
# bdarray(): array island syntax
# name of intermediate result
# destination schema
# destination island type
Output:

```
> curl -X POST -d "bdrel(SELECT * FROM bdcast(bdarray(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:

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

Output:
DATE: [**2976-11-28**] 9:17 PM
CHEST (PORTABLE AP)

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:

```plaintext
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:
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Start</th>
<th>Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigdawg-accumulo-proxy</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-accumulo-master</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-accumulo-tserver0</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-accumulo-zookeeper</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-accumulo-namenode</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-scidb-data</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-postgres-data2</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-postgres-data1</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bigdawg-postgres-catalog</td>
<td>exited</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:
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