Pivotal Summit 2019 Singapore

14 November

Bringing Cloud Databases On-Premises with Greenplum and Kubernetes

Suhail Gulzar
Senior Manager, Data Platform Architecture
Pivotal

Pivotal

Pivotal
Greenplum®

Bringing Cloud Database on-Prem Greenplum for Kubernetes

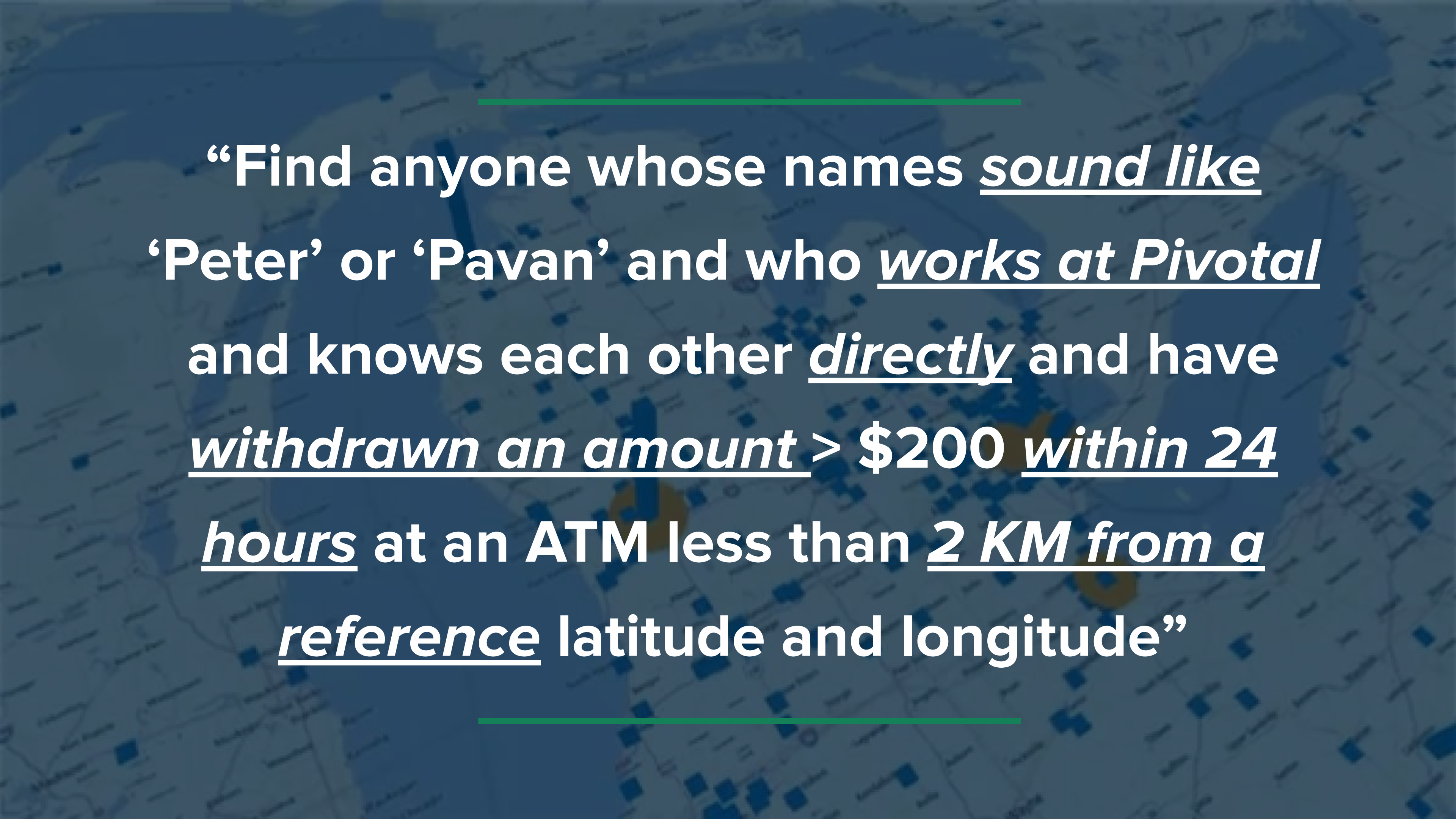
Suhail Gulzar
Data @ Pivotal

Pivotal

A Day in the Life of a Data Scientist

This is a real scenario

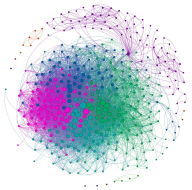
https://github.com/pnagula/Greenplum-Super-Query



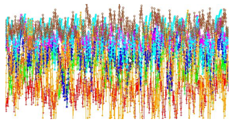
"Find anyone whose names *sound like* 'Peter' or 'Pavan' and who *works at Pivotal* and knows each other *directly* and have *withdrawn an amount* > $200 *within 24 hours* at an ATM less than *2 KM from a reference* latitude and longitude"

An interesting Challenge!!!

Language Analytics
Are these the same words?



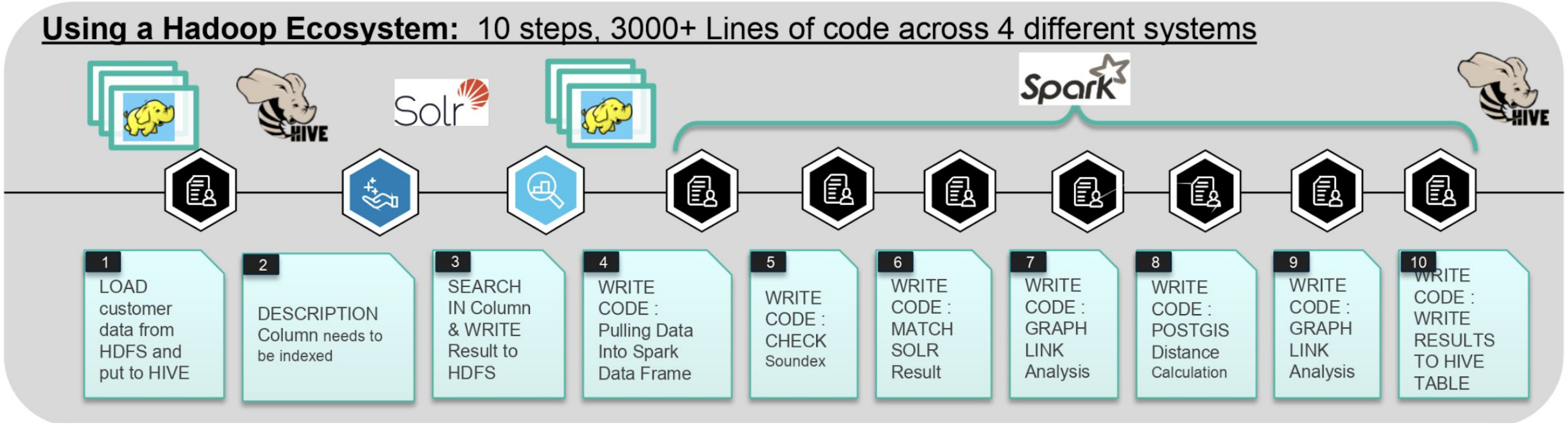
Graph Analytics
Who do they know?



Time Series
When did it happen?

Geospatial Analytics
Where are they?

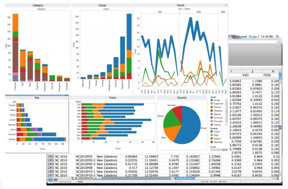
We have Legacy Data Lake/Swamp


Using a Hadoop Ecosystem: 10 steps, 3000+ Lines of code across 4 different systems
HIVE
Solr
Spark
HIVE
1 LOAD customer data from HDFS and put to HIVE
2 DESCRIPTION Column needs to be indexed
3 SEARCH IN Column & WRITE Result to HDFS
4 WRITE CODE : Pulling Data Into Spark Data Frame
5 WRITE CODE : CHECK Soundex
6 WRITE CODE : MATCH SOLR Result
7 WRITE CODE : GRAPH LINK Analysis
8 WRITE CODE : POSTGIS Distance Calculation
9 WRITE CODE : GRAPH LINK Analysis
10 WRITE CODE : WRITE RESULTS TO HIVE TABLE

Fortunately we have Greenplum

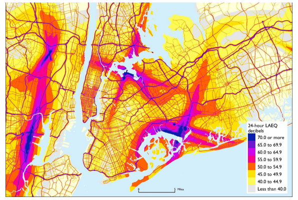
Data Transformation

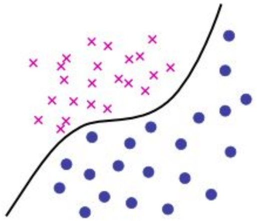
Text



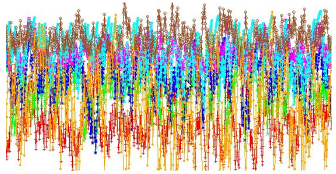
Accelerate
Traditional BI Tools

Geospatial

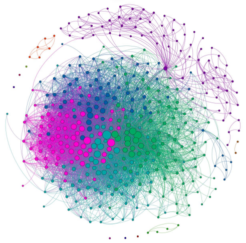




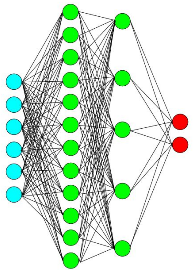
Machine
Learning



Time Series
Analysis



Graph



Deep
Learning

Where should I run this?

Bare-Metal

Private Cloud

vmware vSphere

openstack

Public Cloud

amazon web services

Google Cloud Platform

Microsoft Azure

Greenplum for Kubernetes

Pivotal Container Service

Enterprise & Essentials(OSS K8s)

Google Container Engine

Other Kubernetes (on VMs or not)

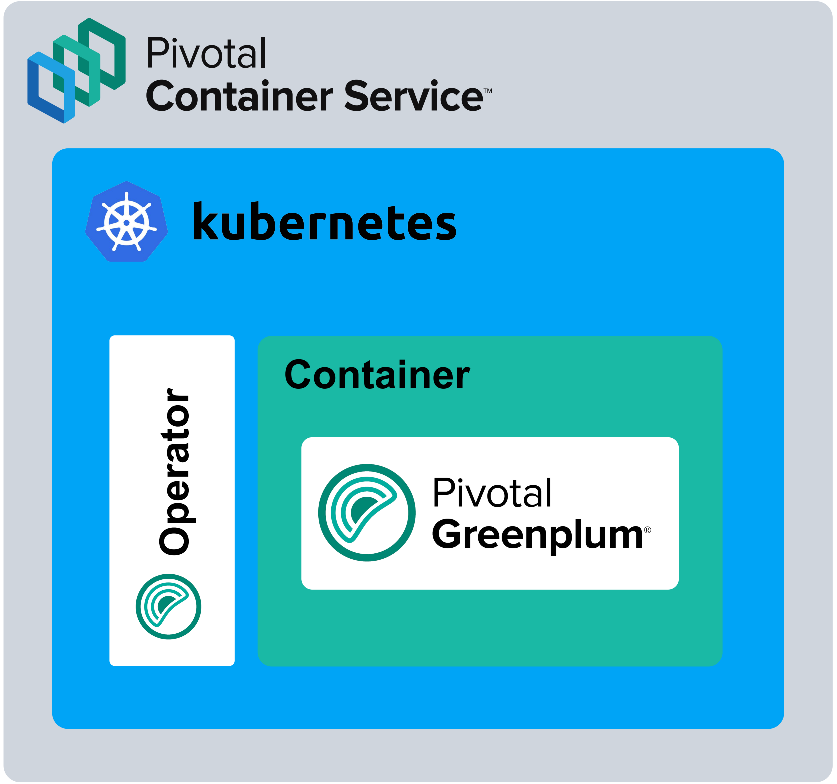
Greenplum Building Blocks

- The most performant way to run Greenplum on premise
- Pivotal Blueprint for Dell reference hardware configs
- Superior price/performance; no expensive proprietary hardware
- Certified and supported by Pivotal

DELL

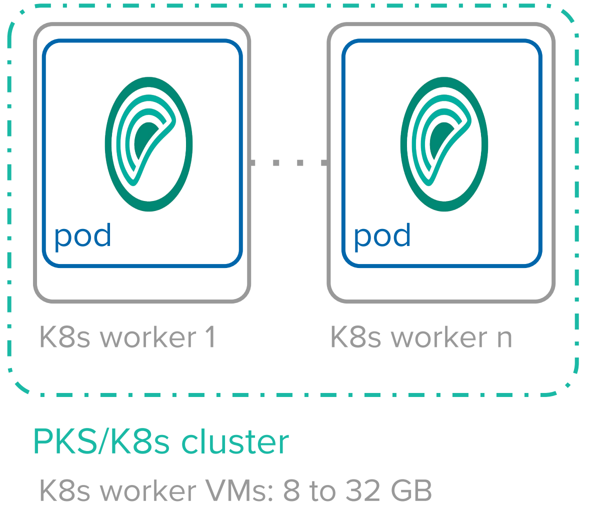
GP4K - Cloud Database On-Premises

- Greenplum is embedded in **containers** for portability and dependency management
- Each container is managed by **Kubernetes** for higher availability & elasticity
- Kubernetes **operator** is used for automation
- **PKS** for multi-cloud and day-2 operations with full-stack support



Deploy Quickly & Easily

- 1 Greenplum Segment =
  Postgres Instance / Pod / Virtual Machine
  (vMotion benefit)
- Local Persistent Volumes
- Consistently Repeatable
- Pre-networked
- Pre-hardened
- Can be deployed as part of an automated pipeline



My friendly Ops Team has done some “One-Time Setup” for me.

K8s Cluster Ready
Operator Ready

- Downloaded Greenplum for K8s
- Uploaded images to registry
- Created K8s cluster
- Deployed GP Operator
- Prepared instance manifest
    - add extensions
    - adjust storage, sizing, etc.

I get to decide what storage to use!

- Kubernetes provides that flexibility
- There are a growing number of storage classes:
  - local for performance
  - remote for flexibility
  - others with features such as dynamic growth
- Users can choose the best storage class for their needs.

Azure Disk

PortworxVolume

VsphereVolume

GCEPersistentDisk

VxFlex OS (formerly ScaleIO)

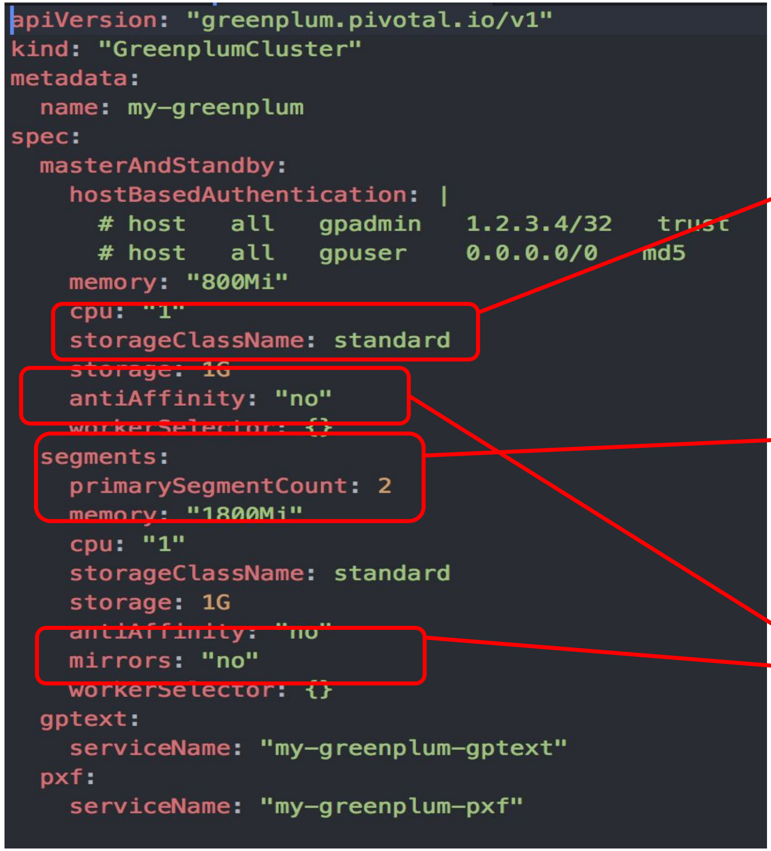
AWSElasticBlockStore

DAS - SSD

DAS - SATA

DriveScale

I get to decide what options to use!



```
apiVersion: "greenplum.pivotal.io/v1"
kind: "GreenplumCluster"
metadata:
  name: my-greenplum
spec:
  masterAndStandby:
    hostBasedAuthentication: |
      # host    all    gpadmin    1.2.3.4/32    trust
      # host    all    gpuser     0.0.0.0/0    md5
    memory: "800Mi"
    cpu: "1"
    storageClassName: standard
    storage: 1G
    antiAffinity: "no"
    workerSelector: {}
  segments:
    primarySegmentCount: 2
    memory: "1800Mi"
    cpu: "1"
    storageClassName: standard
    storage: 1G
    antiAffinity: "no"
    mirrors: "no"
    workerSelector: {}
  gptext:
    serviceName: "my-greenplum-gptext"
  pxf:
    serviceName: "my-greenplum-pxf"
```

For Best Performance:

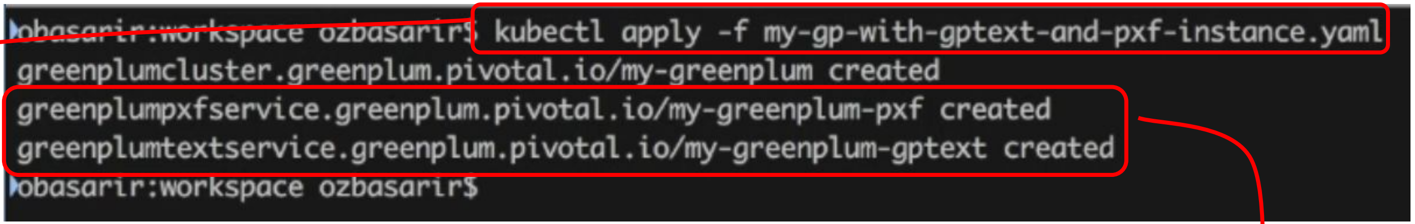
- Backed by a local SSD
- XFS filesystem, using `readahead` cache

- Only 2 Segments to get started
- 1 GB each because we are in Dev.

- No Mirrors we are in Dev.
- AntiAffinity turned off with no mirrors

Same Command

- Initialize Greenplum Workbench
- Update Configuration
- Upgrade Minor Versions
- Apply Patches

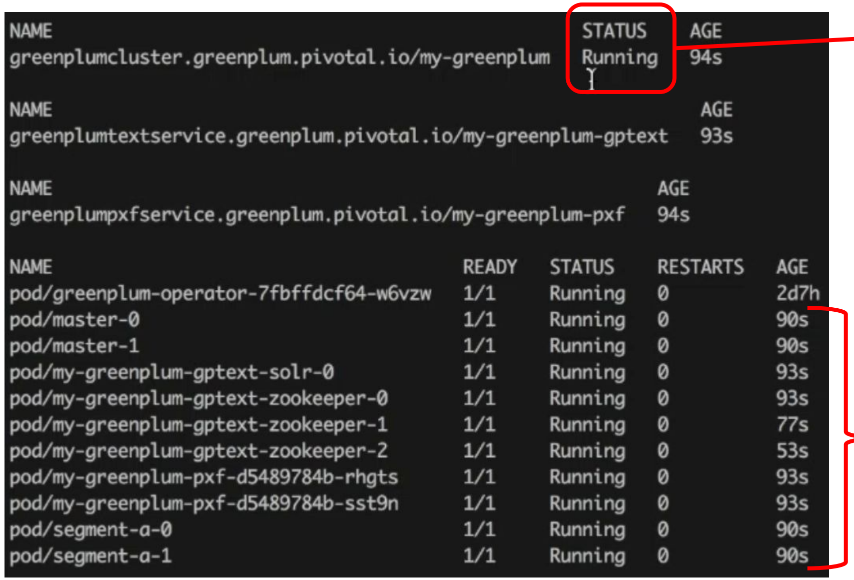


```
obasarir:workspace ozbasarir$ kubectl apply -f my-gp-with-gptext-and-pxf-instance.yaml
greenplumcluster.greenplum.pivotal.io/my-greenplum created
greenplumpxfservice.greenplum.pivotal.io/my-greenplum-pxf created
greenplumtextservice.greenplum.pivotal.io/my-greenplum-gptext created
obasarir:workspace ozbasarir$
```

Kubectl apply -f my-gp.yaml

- Options installed automatically

Ready for User Queries once GP Operator completes



```
NAME                                                STATUS    AGE
greenplumcluster.greenplum.pivotal.io/my-greenplum  Running   94s

NAME                                                         AGE
greenplumtextservice.greenplum.pivotal.io/my-greenplum-gptext   93s

NAME                                                     AGE
greenplumpxfservice.greenplum.pivotal.io/my-greenplum-pxf   94s

NAME                                      READY   STATUS    RESTARTS   AGE
pod/greenplum-operator-7fbffdcf64-w6vzw   1/1     Running   0          2d7h
pod/master-0                              1/1     Running   0          90s
pod/master-1                              1/1     Running   0          90s
pod/my-greenplum-gptext-solr-0            1/1     Running   0          93s
pod/my-greenplum-gptext-zookeeper-0       1/1     Running   0          93s
pod/my-greenplum-gptext-zookeeper-1       1/1     Running   0          77s
pod/my-greenplum-gptext-zookeeper-2       1/1     Running   0          53s
pod/my-greenplum-pxf-d5489784b-rhgts      1/1     Running   0          93s
pod/my-greenplum-pxf-d5489784b-sst9n      1/1     Running   0          93s
pod/segment-a-0                           1/1     Running   0          90s
pod/segment-a-1                           1/1     Running   0          90s
```

- I can start running queries once STATUS switches from PENDING to RUNNING.
- **This setup took 91 Seconds**

- I can see that the Greenplum Operator has created and brought all needed K8s resources to RUNNING state for this GPDB Workbench.
- In case of failures, K8s and the Greenplum Operator work to bring the GPDB Workbench back to its "desired state".

TIme to load up some Data

hadoop PostgreSQL

HIVE MySQL

APACHE HBASE ORACLE

Parquet {JSON} CSV

AVRO Apache orc <xml />

Google Cloud Storage

Aliyun Alibaba Cloud Computing

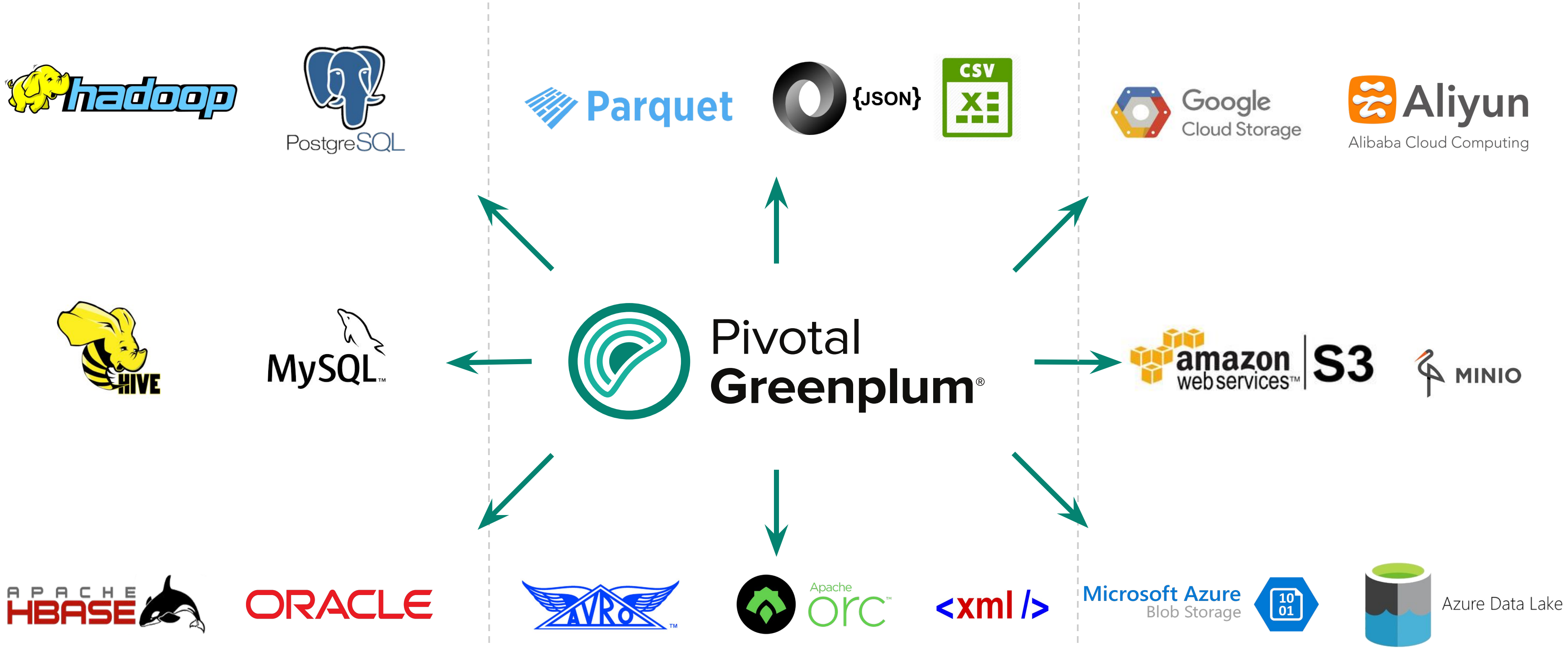
amazon web services S3

MINIO

Microsoft Azure Blob Storage

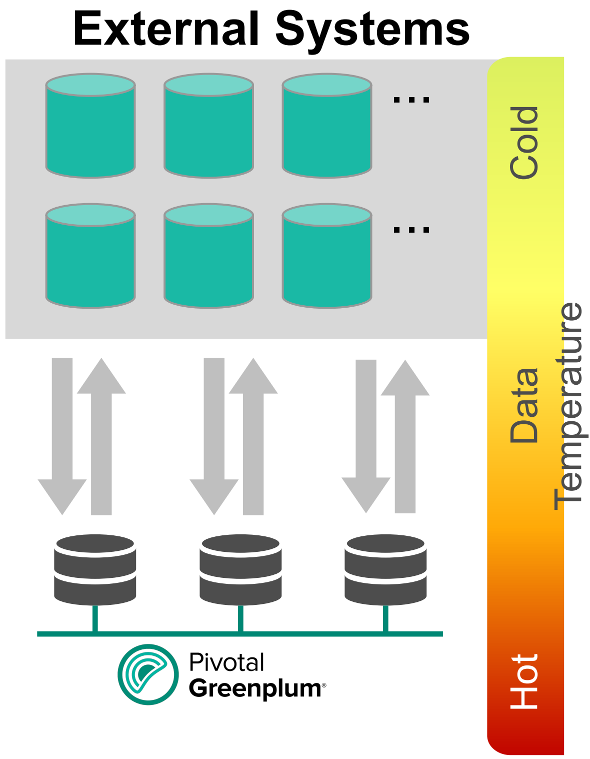
Azure Data Lake

Greenplum can access it all.

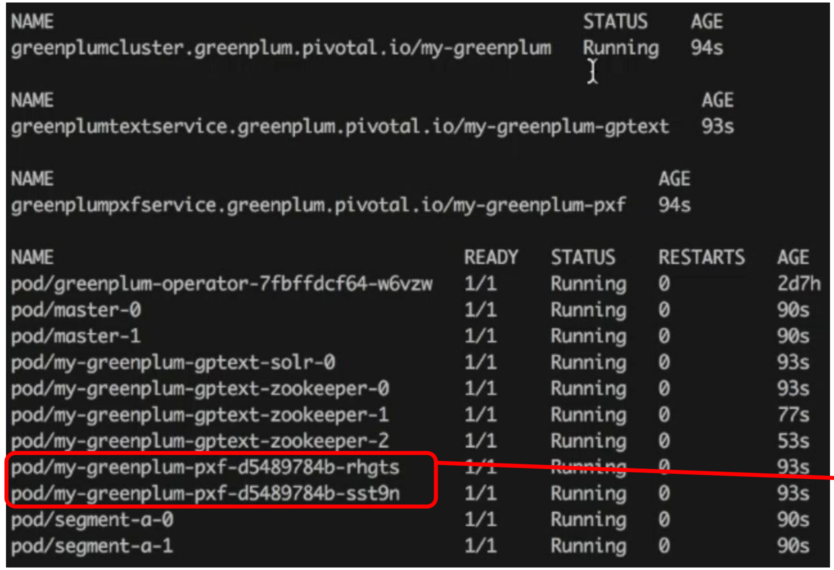


Pivotal Extension Framework (PXF)

- Parallel Access.
- Push Down Processing
- **High Speed (10+ TB/hour) Loading**
- Schema on Read
- Federated Queries
- Standard SQL Interface
- Scale storage independently from compute



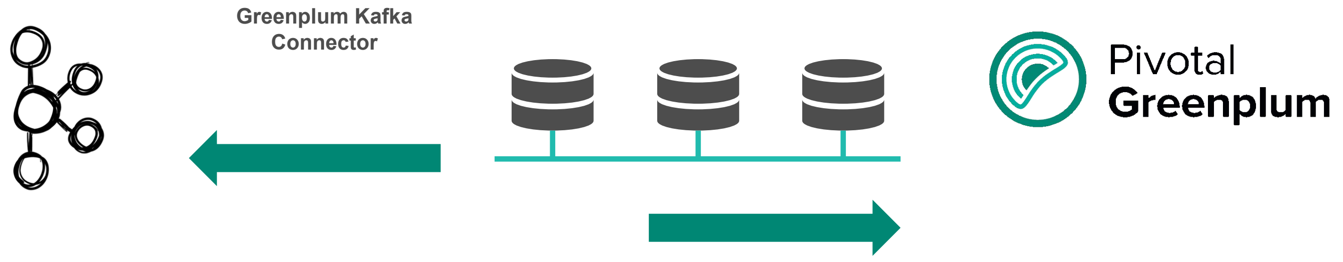
I love my Ops Team - PXF Installed by Default



```
NAME                                             STATUS    AGE
greenplumcluster.greenplum.pivotal.io/my-greenplum   Running   94s

NAME                                                     AGE
greenplumtextservice.greenplum.pivotal.io/my-greenplum-gptext   93s

NAME                                                 AGE
greenplumpxfservice.greenplum.pivotal.io/my-greenplum-pxf   94s

NAME                                         READY   STATUS    RESTARTS   AGE
pod/greenplum-operator-7fbffdcf64-w6vzw      1/1     Running   0          2d7h
pod/master-0                                 1/1     Running   0          90s
pod/master-1                                 1/1     Running   0          90s
pod/my-greenplum-gptext-solr-0               1/1     Running   0          93s
pod/my-greenplum-gptext-zookeeper-0          1/1     Running   0          93s
pod/my-greenplum-gptext-zookeeper-1          1/1     Running   0          77s
pod/my-greenplum-gptext-zookeeper-2          1/1     Running   0          53s
pod/my-greenplum-pxf-d5489784b-rhgts         1/1     Running   0          93s
pod/my-greenplum-pxf-d5489784b-sst9n         1/1     Running   0          93s
pod/segment-a-0                              1/1     Running   0          90s
pod/segment-a-1                              1/1     Running   0          90s
```

- PXF config is setup automatically
- Scale PXF resources independently of GPDB
- We have installed 2 PXF Servers for HA & Perf.

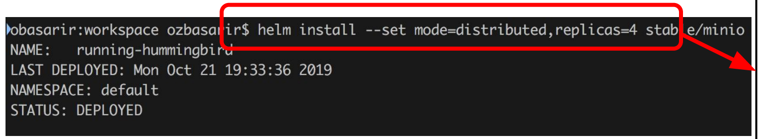
I considered using Kafka



Features:

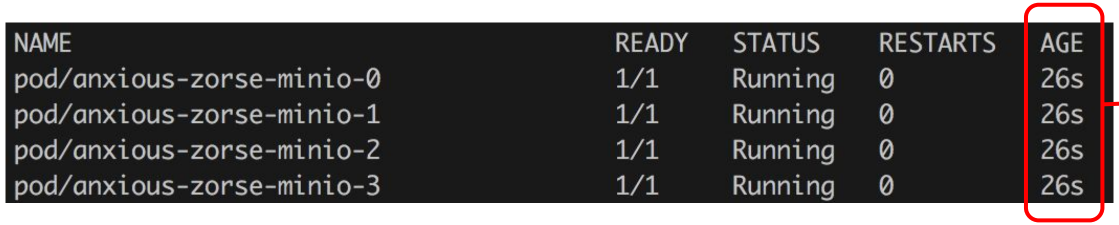
- **Continual** Data Loading
- **Resumable** with Strong **Consistency** Guarantees
- Confluent **Certified**
- Flexible **transformations**
- Automated **aggregations**
- Issue **user defined SQL** on commit

Decided to keep it simple with loading into S3



```
obasarir:workspace ozbasarir$ helm install --set mode=distributed,replicas=4 stable/minio
NAME:   running-hummingbird
LAST DEPLOYED: Mon Oct 21 19:33:36 2019
NAMESPACE: default
STATUS: DEPLOYED
```

- helm install --set mode=distributed,replicas=4 stable/minio
- 4 Nodes of S3 to feed 2 Nodes of GPDB



```
NAME                        READY   STATUS    RESTARTS   AGE
pod/anxious-zorse-minio-0   1/1     Running   0          26s
pod/anxious-zorse-minio-1   1/1     Running   0          26s
pod/anxious-zorse-minio-2   1/1     Running   0          26s
pod/anxious-zorse-minio-3   1/1     Running   0          26s
```

- Started running in under 30s

“Find anyone whose names *sound like* ‘Peter’ or ‘Pavan’ and who *works at Pivotal*

GPText and Greenplum

Extract and Transform

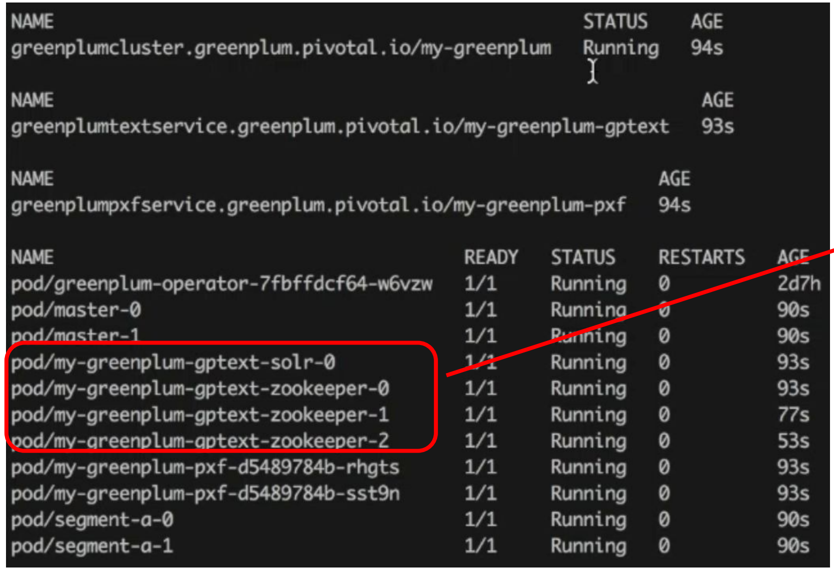
- Fast text extraction, indexing/search
- Open source analytics with MPP processing
- Index/store metadata only, avoid data ETL
- Search-engine like syntax
- Better matching for more relevant results
- Many sources and formats, w/o complexity

Explore and Analyze

- Part of Speech Detection
- Named Entity Recognition
- Categorization (via MADLib)
- Topic Modeling (via MADLib)
- Classification/Sentiment (via MADlib, Python, R libraries)

Identify language that signals interesting behaviors and events for the use case

I really love my Ops - GPText Installed by Default



```
NAME                                              STATUS    AGE
greenplumcluster.greenplum.pivotal.io/my-greenplum   Running   94s

NAME                                                       AGE
greenplumtextservice.greenplum.pivotal.io/my-greenplum-gptext   93s

NAME                                                 AGE
greenplumpxfservice.greenplum.pivotal.io/my-greenplum-pxf   94s

NAME                                       READY   STATUS    RESTARTS   AGE
pod/greenplum-operator-7fbffdcf64-w6vzw    1/1     Running   0          2d7h
pod/master-0                               1/1     Running   0          90s
pod/master-1                               1/1     Running   0          90s
pod/my-greenplum-gptext-solr-0             1/1     Running   0          93s
pod/my-greenplum-gptext-zookeeper-0        1/1     Running   0          93s
pod/my-greenplum-gptext-zookeeper-1        1/1     Running   0          77s
pod/my-greenplum-gptext-zookeeper-2        1/1     Running   0          53s
pod/my-greenplum-pxf-d5489784b-rhgts       1/1     Running   0          93s
pod/my-greenplum-pxf-d5489784b-sst9n       1/1     Running   0          93s
pod/segment-a-0                            1/1     Running   0          90s
pod/segment-a-1                            1/1     Running   0          90s
```

- Installed by automatically
- Scale GPText resources independently of GPDB
- Running 3 instances.

Find anyone whose names sound like 'Peter' or 'Pavan'and who work at 'Pivotal' and know each other 'directly' and have withdrawn an amount > $200 within 24 hours at an ATM less than 2 KM from reference latitude and longitude.

Greenplum Fuzzy String Match function **Soundex()** to know if people name sounds like 'Pavan' or 'Peter'

GPText.search() function is used to know if both people work at 'Pivotal'

```
drop function if exists get_people(text,text,integer,integer,float,float);
CREATE FUNCTION get_people(text,text,integer,integer,float,float) RETURNS integer
AS $$
declare
linkchk integer; v1 record; v2 record;
begin
 execute 'truncate table results;';
 for v1 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($1) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4

and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
   for v2 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($2) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4
    and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
     execute 'DROP TABLE IF EXISTS out, out_summary;';
     execute  'SELECT madlib.graph_bfs(''people'',''id'',''links'',NULL,'||v1.id||',''out'');'  ;
     select 1 into linkchk from out where dist=1 and id=v2.id;
     if linkchk is not null  then
          insert into results values (v1.id,v1.firstname,v1.lastname,v1.amount,v1.tran_date,v1.lat,v1.lng,v1.address,v1.description,v1.score);
          insert into results values (v2.id,v2.firstname,v2.lastname,v2.amount,v2.tran_date,v2.lat,v2.lng,v2.address,v2.description,v2.score);
      end if;
    end loop;
 end loop;
 return 0;
end
$$ LANGUAGE plpgsql;
--        person1 , person 2, amount, duration in hours, longtitude, latitude (in question)
select get_people('Pavan','Peter',200,24,103.912680, 1.309432) ;
```

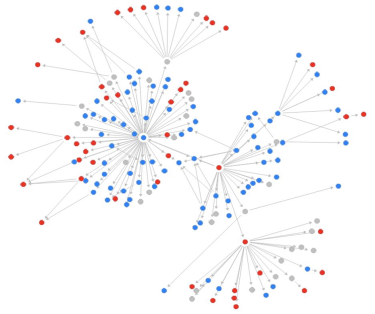
Knows each other *directly*

Graph Analytics - finding networks.

Social Network

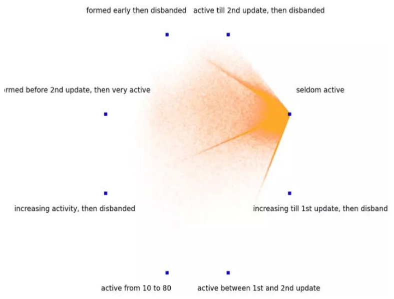
* Grandjean, M. (2016)

Epidemiology



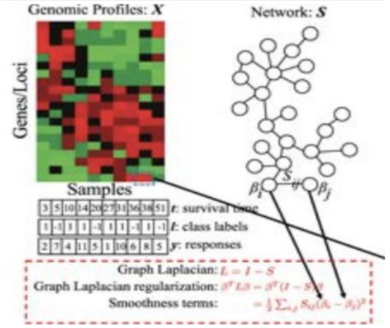
* http://www.netminer.com/community

MMO Role-Playing Game



* www.researchgate.net

Chemistry

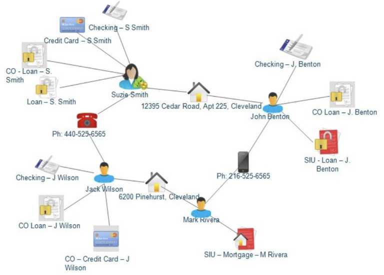


* https://www.nature.com/articles/

Bank Risk

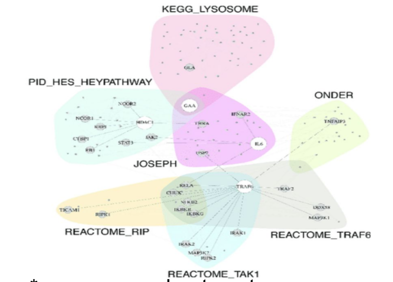
* https://cambridge-intelligence.com

1st Party Fraud



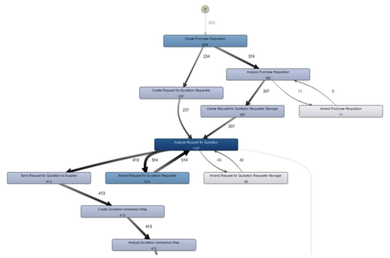
* www.infoglide.com

Gene



* www.researchgate.net

Manufacturing



* https://blog.trifinance.com

Find anyone whose names sound like ‘Peter’ or ‘Pavan’and who work at ‘Pivotal’ and know each other ‘directly’ and have withdrawn an amount > $200 within 24 hours at an ATM less than 2 KM from reference latitude and longitude.

```
drop function if exists get_people(text,text,integer,integer,float,float);
CREATE FUNCTION get_people(text,text,integer,integer,float,float) RETURNS integer
AS $$
declare
linkchk integer; v1 record; v2 record;
begin
 execute 'truncate table results;';
 for v1 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($1) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4

and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
   for v2 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($2) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4
    and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
     execute 'DROP TABLE IF EXISTS out, out_summary;';
     execute  'SELECT madlib.graph_bfs(''people'',''id'',''links'',NULL,'||v1.id||',''out'');'  ;
     select 1 into linkchk from out where dist=1 and id=v2.id;
     if linkchk is not null  then
          insert into results values (v1.id,v1.firstname,v1.lastname,v1.amount,v1.tran_date,v1.lat,v1.lng,v1.address,v1.description,v1.score);
          insert into results values (v2.id,v2.firstname,v2.lastname,v2.amount,v2.tran_date,v2.lat,v2.lng,v2.address,v2.description,v2.score);
      end if;
    end loop;
 end loop;
 return 0;
end
$$ LANGUAGE plpgsql;
--        person1 , person 2, amount, duration in hours, longtitude, latitude (in question)
select get_people('Pavan','Peter',200,24,103.912680, 1.309432) ;
```

Greenplum and **Apache MADlib BFS** search to know if there are direct or indirect links between people

Disaster strikes - a Node Fails!!

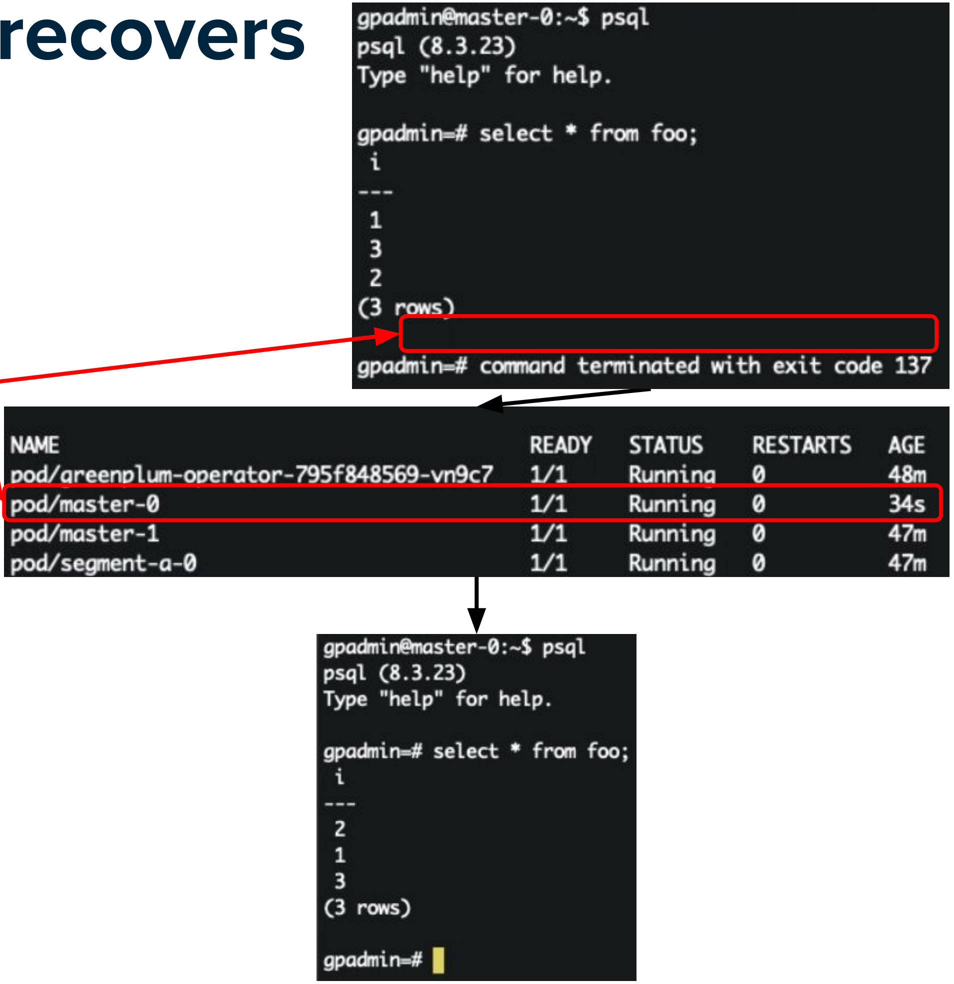
Node Fails - GPDB Auto-recovers

No manual recovery needed;
Just re-run the query!

Master dies and is recovered in 34s.
Same process applies to segments.

Even if its host dies, the master (or segment) will recover on another host because of compute-storage separation.

If you use remote storage then mirrors are not required for auto-recovery.



```
gpadmin@master-0:~$ psql
psql (8.3.23)
Type "help" for help.

gpadmin=# select * from foo;
 i
---
 1
 3
 2
(3 rows)

gpadmin=# command terminated with exit code 137
```

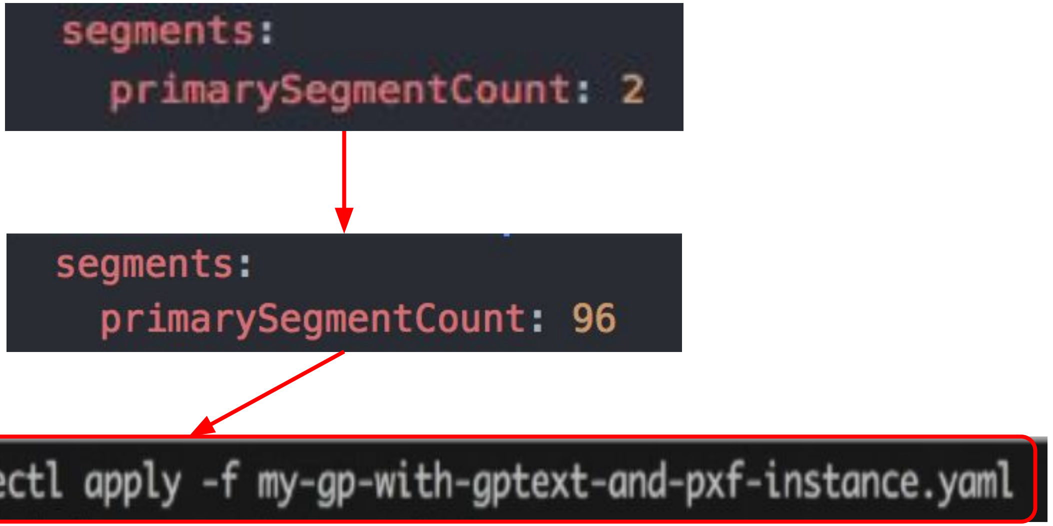
```
NAME                                    READY   STATUS    RESTARTS   AGE
pod/greenplum-operator-795f848569-vn9c7 1/1     Running   0          48m
pod/master-0                            1/1     Running   0          34s
pod/master-1                            1/1     Running   0          47m
pod/segment-a-0                         1/1     Running   0          47m
```

```
gpadmin@master-0:~$ psql
psql (8.3.23)
Type "help" for help.

gpadmin=# select * from foo;
 i
---
 2
 1
 3
(3 rows)

gpadmin=#
```

More Data Comes In - Expand GPDB Cluster

- I edit the yaml
- I resubmit the kubectl
- Cluster expands
- GPDB - Autoexpands

- Tested out to 128 Segments
- Linear scaling
- Similar performance to Bare Metal



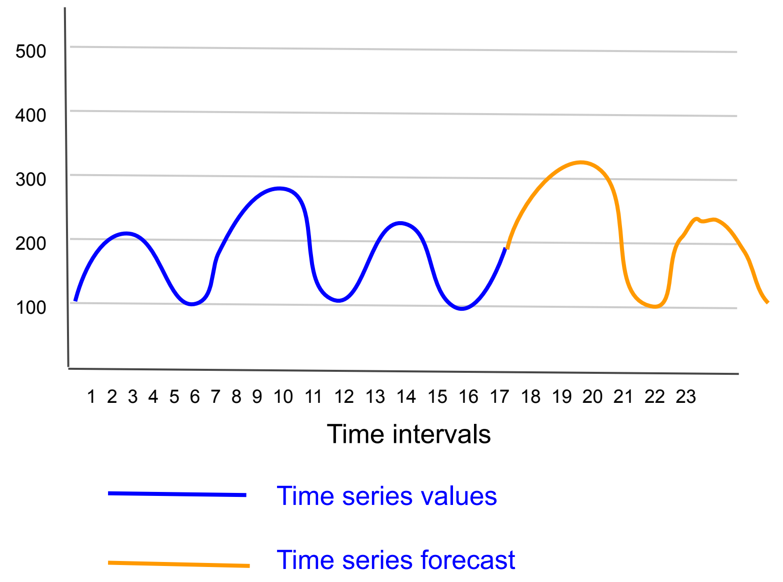
```
segments:
  primarySegmentCount: 2
```

```
segments:
  primarySegmentCount: 96
```

```
obasarir:workspace ozbasarir$ kubectl apply -f my-gp-with-gptext-and-pxf-instance.yaml
```

withdrawn an amount **> $200** ***within 24 hours***

Greenplum is a Time Series Database



Time Series Data

- Series of data points indexed in time order.
- Primarily inserts with the recent time interval.
- Commonly equally spaced time intervals

Time series analysis

- Methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data.

Time series forecasting

- Use of a model to predict future values based on previously observed values

Find anyone whose names sound like ‘Peter’ or ‘Pavan’and who work at ‘Pivotal’ and know each other ‘directly’ and have withdrawn an amount > $200 within 24 hours at an ATM less than 2 KM from reference latitude and longitude.

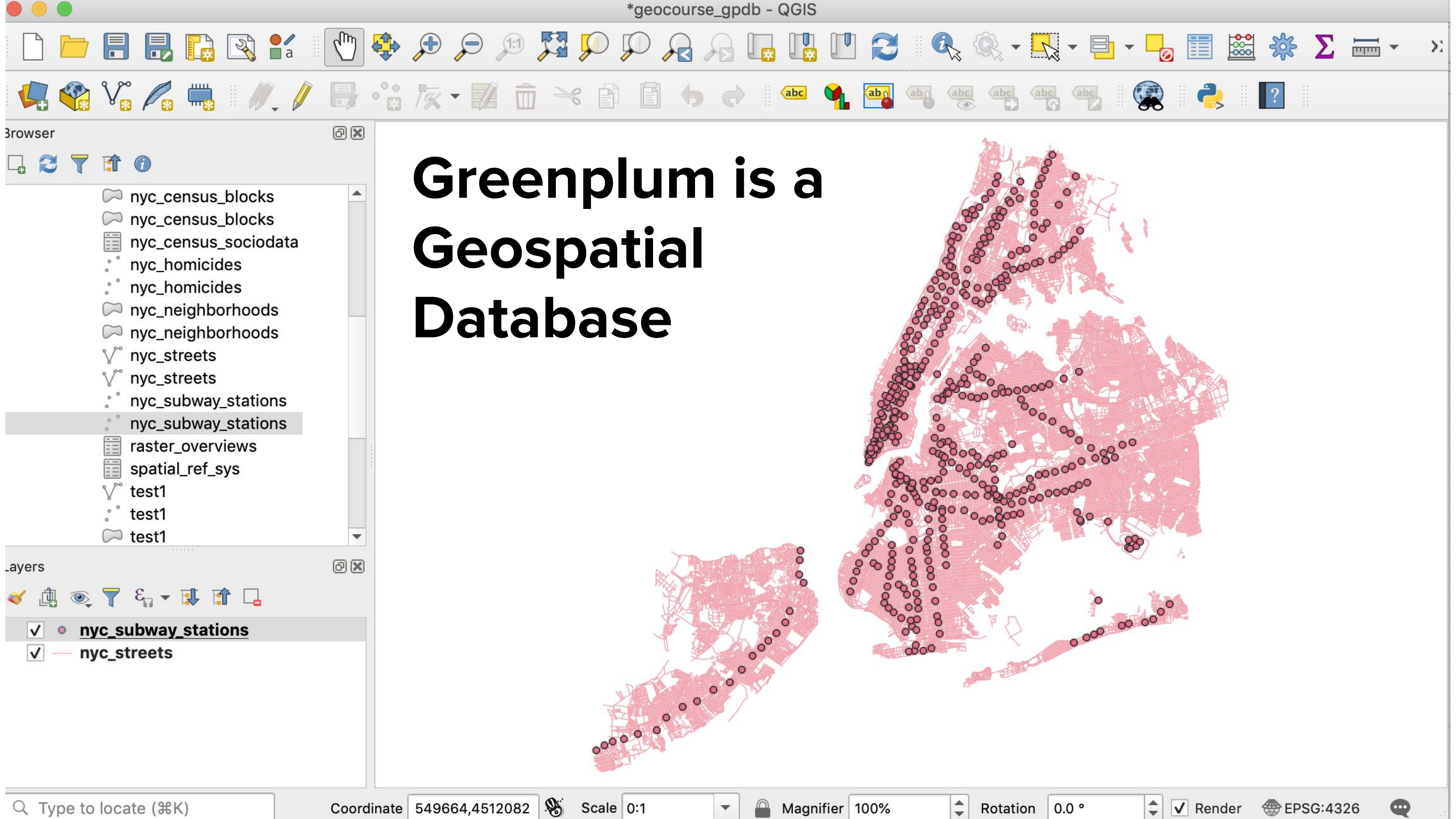
```
drop function if exists get_people(text,text,integer,integer,float,float);
CREATE FUNCTION get_people(text,text,integer,integer,float,float) RETURNS integer
AS $$
declare
linkchk integer; v1 record; v2 record;
begin
 execute 'truncate table results;';
 for v1 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($1) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4

and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
   for v2 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($2) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4
    and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
     execute 'DROP TABLE IF EXISTS out, out_summary;';
     execute  'SELECT madlib.graph_bfs(''people'',''id'',''links'',NULL,'||v1.id||',''out'');'  ;
     select 1 into linkchk from out where dist=1 and id=v2.id;
     if linkchk is not null  then
          insert into results values (v1.id,v1.firstname,v1.lastname,v1.amount,v1.tran_date,v1.lat,v1.lng,v1.address,v1.description,v1.score);
          insert into results values (v2.id,v2.firstname,v2.lastname,v2.amount,v2.tran_date,v2.lat,v2.lng,v2.address,v2.description,v2.score);
      end if;
    end loop;
 end loop;
 return 0;
end
$$ LANGUAGE plpgsql;
--          person1 , person 2, amount, duration in hours, longtitude, latitude (in question)
select get_people('Pavan','Peter',200,24,103.912680, 1.309432) ;
```

Amount > $200

Greenplum **Time functions** to calculate difference in amount withdrawn time < 24 hours

ATM less than *2 KM from a reference* latitude and longitude”

Greenplum is a Geospatial Database



*geocourse_gpdb - QGIS

Browser

- nyc_census_blocks
- nyc_census_blocks
- nyc_census_sociodata
- nyc_homicides
- nyc_homicides
- nyc_neighborhoods
- nyc_neighborhoods
- nyc_streets
- nyc_streets
- nyc_subway_stations
- nyc_subway_stations
- raster_overviews
- spatial_ref_sys
- test1
- test1
- test1

Layers

- ✓ nyc_subway_stations
- ✓ nyc_streets

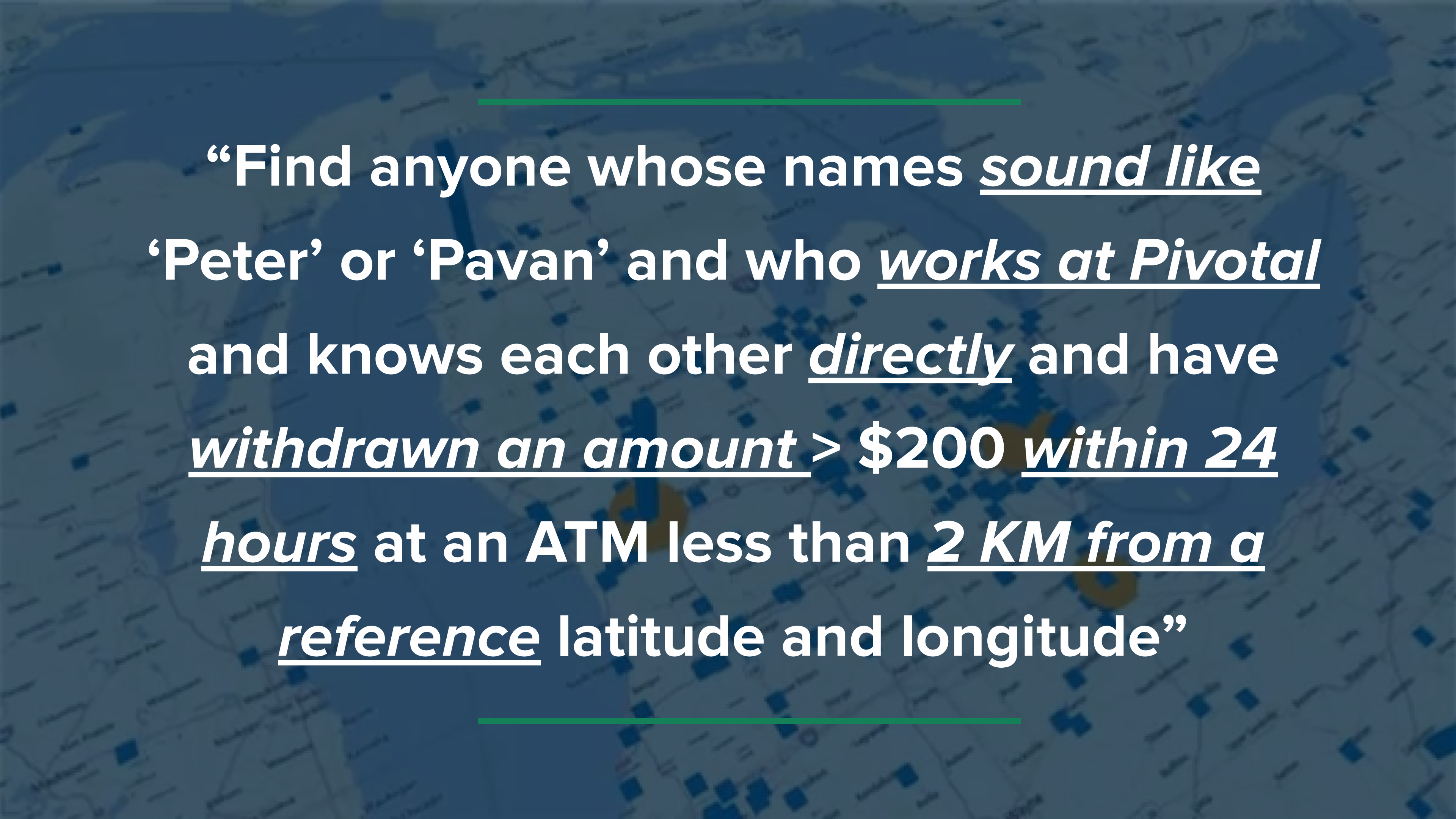
Type to locate (⌘K)

Coordinate 549664,4512082 Scale 0:1 Magnifier 100% Rotation 0.0 ° ✓ Render EPSG:4326

Find anyone whose names sound like ‘Peter’ or ‘Pavan’and who work at ‘Pivotal’ and know each other ‘directly’ and have withdrawn an amount > $200 within 24 hours at an ATM less than 2 KM from reference latitude and longitude.

```
drop function if exists get_people(text,text,integer,integer,float,float);
CREATE FUNCTION get_people(text,text,integer,integer,float,float) RETURNS integer
AS $$
declare
linkchk integer; v1 record; v2 record;
begin
 execute 'truncate table results;';
 for v1 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($1) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4

and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
   for v2 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
    (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
     WHERE (q.id::integer) = w.id order by 2 desc) d
     where soundex(firstname)=soundex($2) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4
    and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
     execute 'DROP TABLE IF EXISTS out, out_summary;';
     execute  'SELECT madlib.graph_bfs(''people'',''id'',''links'',NULL,'||v1.id||',''out'');'  ;
     select 1 into linkchk from out where dist=1 and id=v2.id;
     if linkchk is not null  then
          insert into results values (v1.id,v1.firstname,v1.lastname,v1.amount,v1.tran_date,v1.lat,v1.lng,v1.address,v1.description,v1.score);
          insert into results values (v2.id,v2.firstname,v2.lastname,v2.amount,v2.tran_date,v2.lat,v2.lng,v2.address,v2.description,v2.score);
      end if;
    end loop;
 end loop;
 return 0;
end
$$ LANGUAGE plpgsql;
--        person1 , person 2, amount, duration in hours, longtitude, latitude (in question)
select get_people('Pavan','Peter',200,24,103.912680, 1.309432) ;
```

Greenplum POSTGIS functions **st_distance_sphere() and st_makepoint()** calculate distance between ATM location and reference lat ,long < 2 KM



"Find anyone whose names *sound like* 'Peter' or 'Pavan' and who *works at Pivotal* and knows each other *directly* and have *withdrawn an amount* > $200 *within 24 hours* at an ATM less than *2 KM from a reference* latitude and longitude"

Find anyone whose names sound like 'Peter' or 'Pavan'and who work at 'Pivotal' and know each other 'directly' and have withdrawn an amount > $200 within 24 hours at an ATM less than 2 KM from reference latitude and longitude.

```
drop function if exists get_people(text,text,integer,integer,float,float);
CREATE FUNCTION get_people(text,text,integer,integer,float,float) RETURNS integer
AS $$
declare
linkchk integer; v1 record; v2 record;
begin
 execute 'truncate table results;';
 for v1 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
   (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
    WHERE (q.id::integer) = w.id order by 2 desc) d
    where soundex(firstname)=soundex($1) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4

and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
   for v2 in select distinct a.id,a.firstname,a.lastname,amount,tran_date,c.lat,c.lng,address,a.description,d.score from people a,transactions b,location c,
   (SELECT w.id, q.score FROM people w, gptext.search(TABLE(SELECT 1 SCATTER BY 1), 'gpadmin.public.people' , 'Pivotal', null) q
    WHERE (q.id::integer) = w.id order by 2 desc) d
    where soundex(firstname)=soundex($2) and a.id=b.id and amount > $3 and (extract(epoch from tran_date) - extract(epoch from now()))/3600 < $4
   and st_distance_sphere(st_makepoint($5, $6),st_makepoint(c.lng, c.lat))/1000.0 <= 2.0 and b.locid=c.locid and a.id=d.id
 loop
    execute 'DROP TABLE IF EXISTS out, out_summary;';
    execute  'SELECT madlib.graph_bfs(''people'',''id'',''links'',NULL,'||v1.id||',''out'');'  ;
    select 1 into linkchk from out where dist=1 and id=v2.id;
    if linkchk is not null  then
         insert into results values (v1.id,v1.firstname,v1.lastname,v1.amount,v1.tran_date,v1.lat,v1.lng,v1.address,v1.description,v1.score);
         insert into results values (v2.id,v2.firstname,v2.lastname,v2.amount,v2.tran_date,v2.lat,v2.lng,v2.address,v2.description,v2.score);
     end if;
   end loop;
 end loop;
 return 0;
end
$$ LANGUAGE plpgsql;
--        person1 , person 2, amount, duration in hours, longtitude, latitude (in question)
select get_people('Pavan','Peter',200,24,103.912680, 1.309432) ;
```

Greenplum Fuzzy String Match function **Soundex()** to know if people name sounds like 'Pavan' or 'Peter'

GPText.search() function is used to know if both people work at 'Pivotal'

Amount > $200

Greenplum and **Apache MADlib BFS** search to know if there are direct or indirect links between people

Greenplum **Time functions** to calculate difference in amount withdrawn time < 24 hours

Greenplum POSTGIS functions **st_distance_sphere() and st_makepoint()** calculate distance between ATM location and reference lat ,long < 2 KM

If I had to go into Production - Not Today :)

Real Time Scoring For Apache MADlib

Single command to deploy a MADlib trained model from Pivotal Greenplum / Postgres to Docker, PCF or Kubernetes

```
$ madlib --deploy
```

User Operations

Data Platform Automated Operations

1. **ML Training**

   Train models in Pivotal Greenplum using Apache MADlib

2. **madlib --deploy**

   Auto-deploy model from Greenplum to Docker, PCF or Kubernetes

3. **Docker Build**

   Build docker container with optimized PostgreSQL and MADlib

4. **Synchronize Model**

   Extract model and feature components from Pivotal Greenplum and deploy to container

5. **Load Model**

   Load model and feature components into containerized PostgresSQL and MADlib engine

6. **Deploy ML Engine**

   Deploy docker container as ML REST end point in target PKS environment

Pivotal

Release Compute Resources When Done

Release and Retain State and Data

```
kubectl delete -f my-gp-with-gptext-and-pxf-instance.yaml
```

Patch to a new version

```
kubectl delete -f my-gp-with-gptext-and-pxf-instance.yaml
kubectl apply -f my-gp-with-gptext-and-pxf-instance.yaml
```

Drop Data (Everything gone)

```
kubectl delete pvc --all
```

I thought this was going to take:

- 4 weeks to provision an environment
- 2 Weeks to get a landing zone
- 3 Days to load data
- 1 Week to code 3000 lines of Hadoop Code

Instead it is 4PM and I am off to the gym.