Pgbench on PolarDB-Oracle（PolarDB-Postgresql)

今回はPgbenchのベンチマークテストをご紹介、テスト対象はPolarDB-Oracleです（PolarDB-Postgresqlも同じテスト方法で実行します)

概要

本文章はPgbenchでPolarDB-Oracleをテストする方法をご紹介します。

1 Pgbench環境を準備する

2 PolarDB-Oracleを性能測定する

1 Pgbench環境を準備する

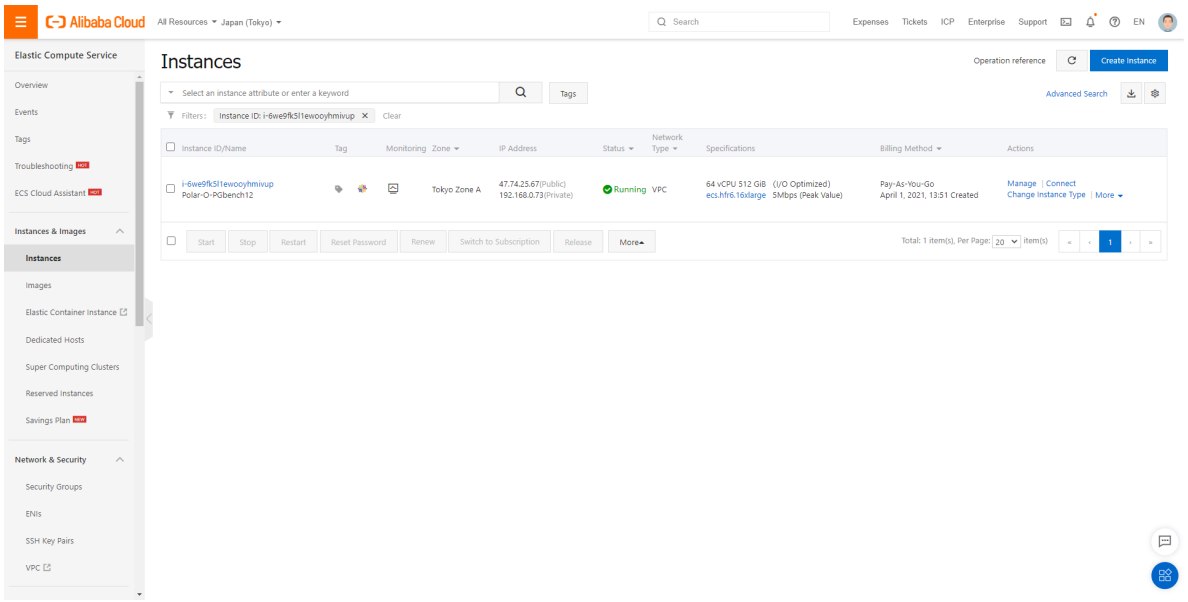
Pgbenchとは

Pgbench はPostgreSQLのベンチマークテストに用いられるプログラムです。

ECSでPgbench環境を用意する

1) ECSインスタンスを作成する

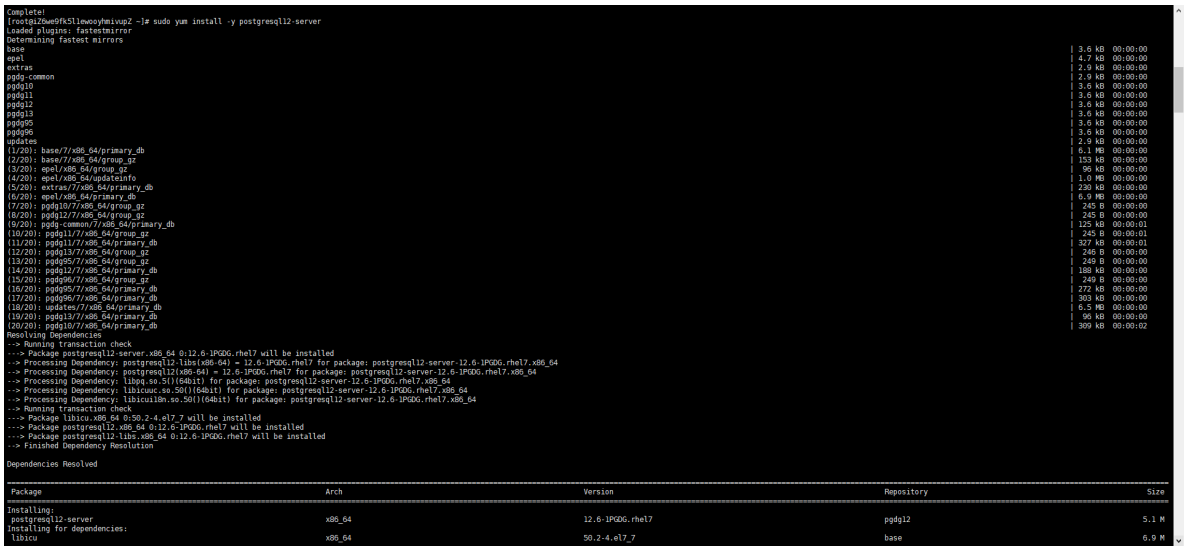
```
ECS:
Specifications: 64 vCPU 512 GiB ecs.r5.16xlarge
OS: CentOS  7.7 64-bit
```

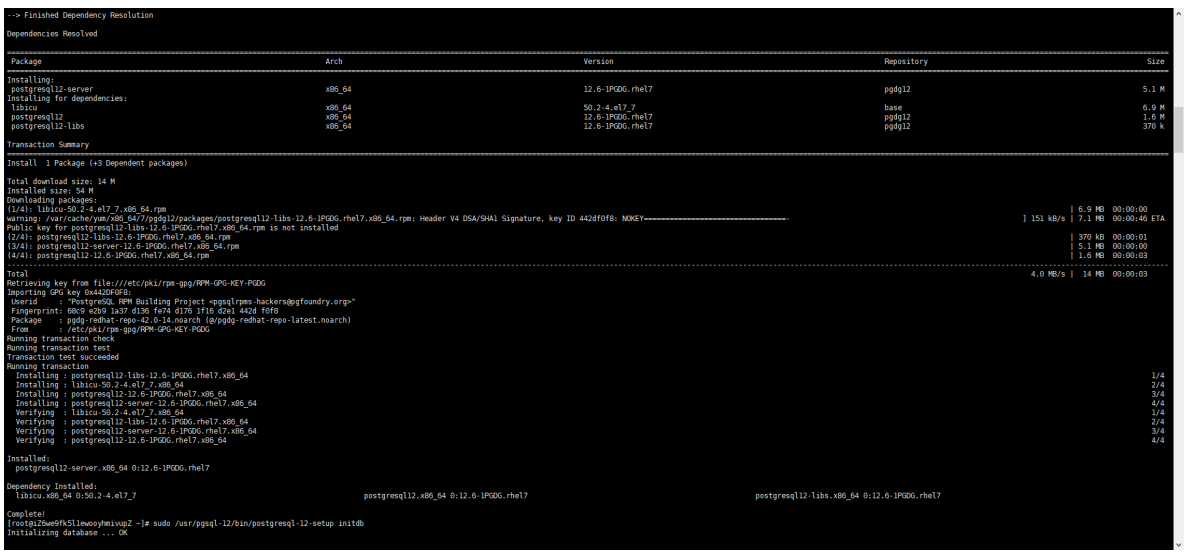


2) ECSでPgbench環境をインストールする

①下記コマンドを実行し、ECSでpostgresql-12をインストールする

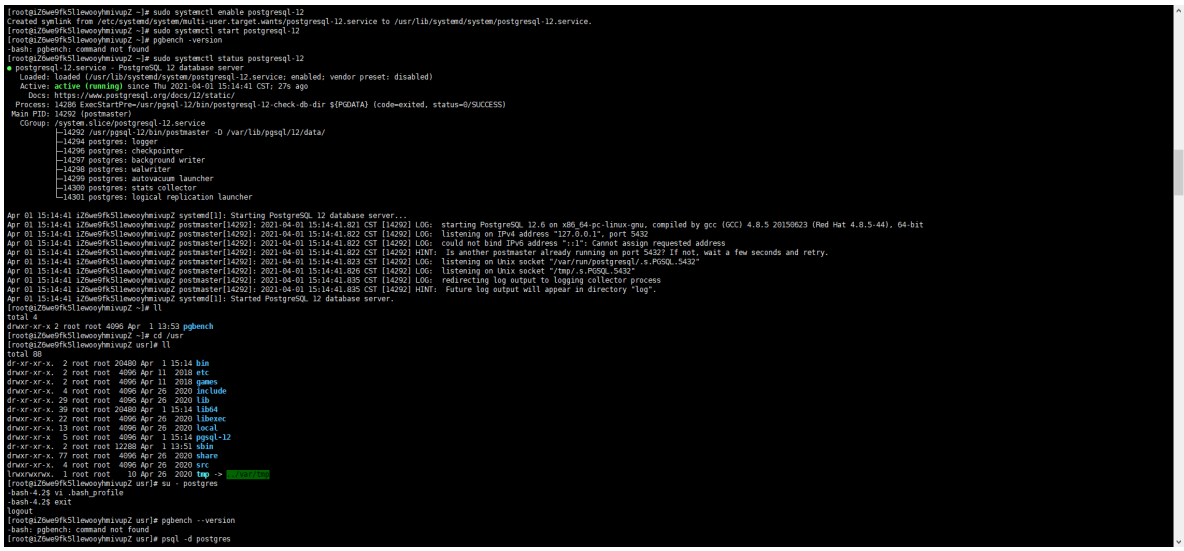
```
# Install the repository RPM:
sudo yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm

# Install PostgreSQL:
sudo yum install -y postgresql12-server

# Optionally initialize the database and enable automatic start:
sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
```





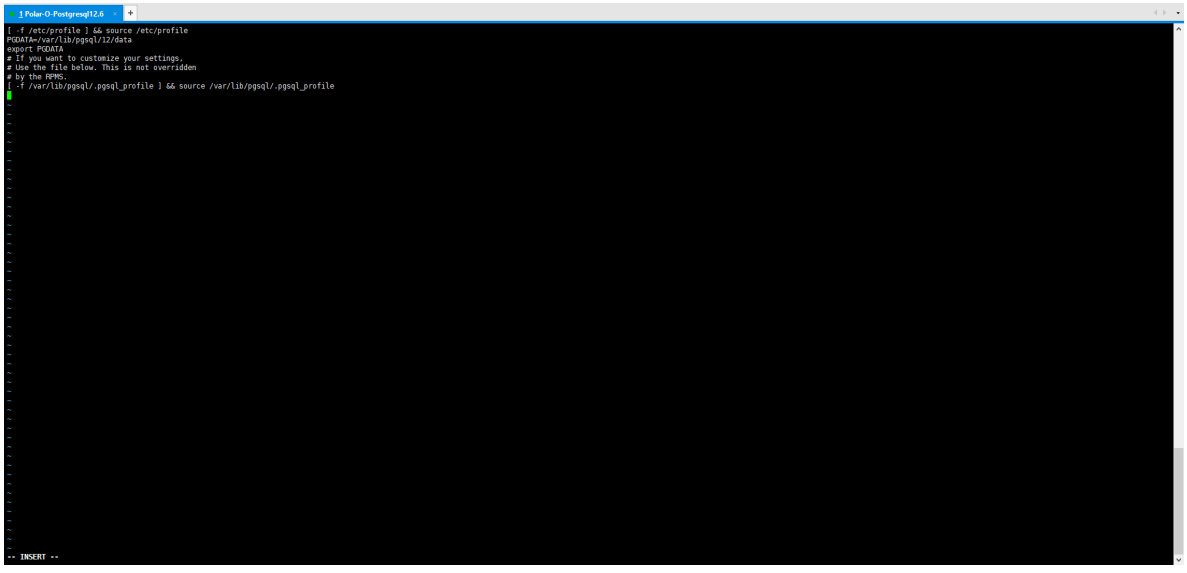
②Postgresql-12を起動する

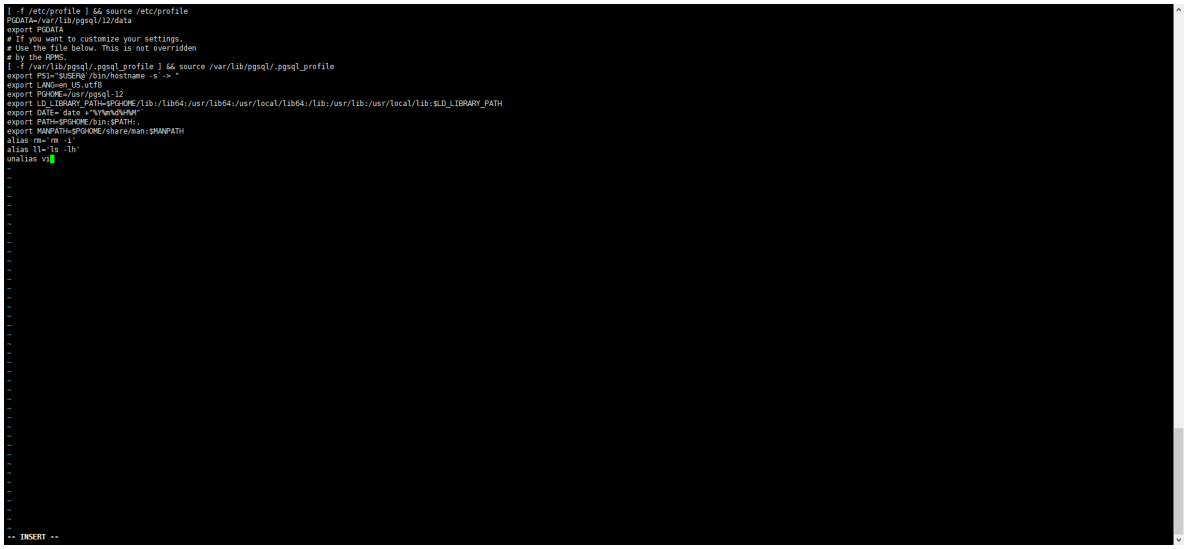
```
sudo systemctl enable postgresql-12
sudo systemctl start postgresql-12
sudo systemctl status postgresql-12
```

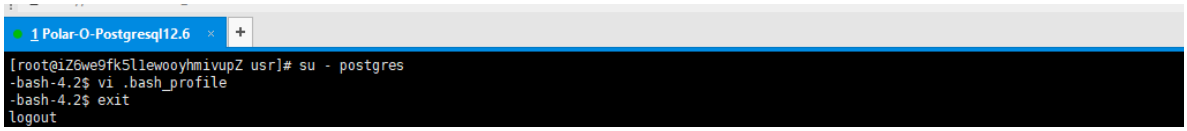


③bash_profileを設定する

```
su - postgres
vi .bash_profile
export PS1="$USER@`/bin/hostname -s`-> "
export LANG=en_US.utf8
export PGHOME=/usr/pgsql-12
export
LD_LIBRARY_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/us
r/local/lib:$LD_LIBRARY_PATH
export DATE=`date +"%Y%m%d%H%M"`
export PATH=$PGHOME/bin:$PATH:.
export MANPATH=$PGHOME/share/man:$MANPATH
alias rm='rm -i'
alias ll='ls -lh'
unalias vi
```







④Postgresqlはデフォルトで/usr/pgsql-12/binにインストールされている、Pathを追加する

```
[root@iZ6we9fk5l1ewooyhmivupZ /]# /etc/profile
-bash: /etc/profile: Permission denied
[root@iZ6we9fk5l1ewooyhmivupZ /]# chmod 755 /etc/profile
[root@iZ6we9fk5l1ewooyhmivupZ /]# /etc/profile
[root@iZ6we9fk5l1ewooyhmivupZ /]# export PATH=$PATH:/usr/pgsql-12/bin
[root@iZ6we9fk5l1ewooyhmivupZ /]# pgbench --version
pgbench (PostgreSQL) 12.6
```

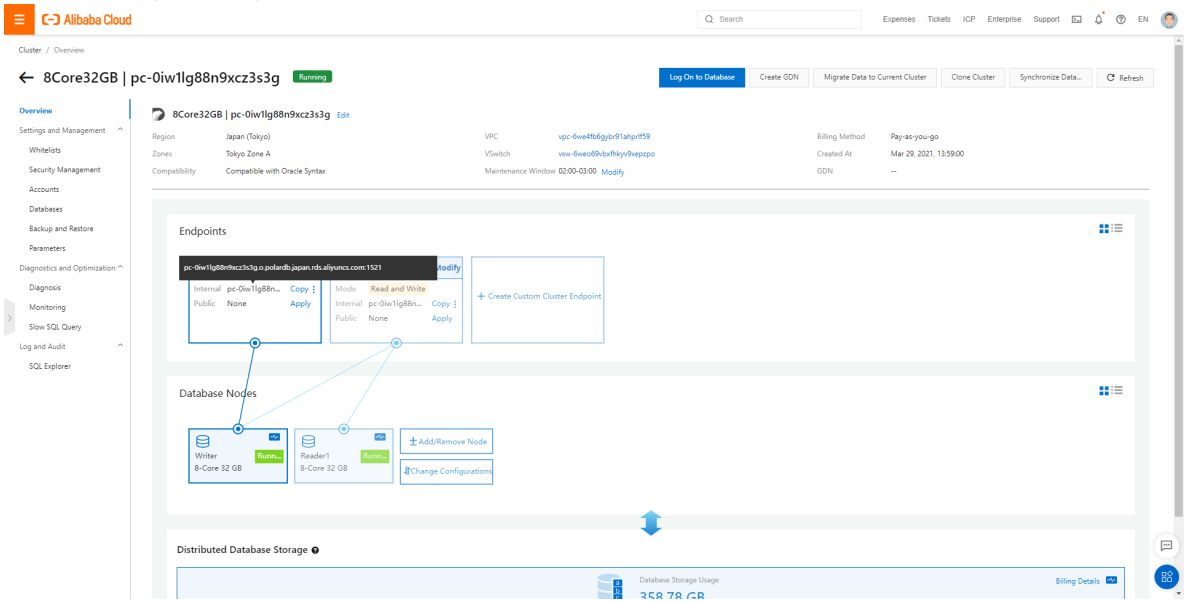
ここまでPgbenchがインストール完了しました

下記のサイトをご参照ください
https://help.aliyun.com/document_detail/118338.html
https://www.postgresql.org/download/linux/redhat/

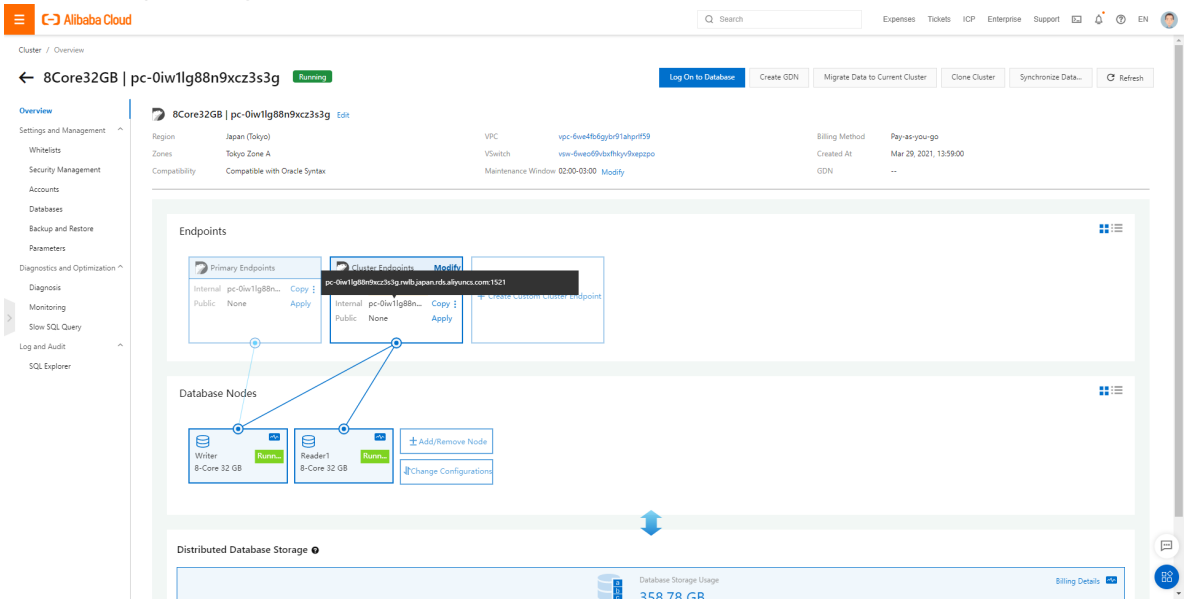
2 PolarDB-Oracleを性能測定する

1) PolarDB-Oracleインスタンスを作成する

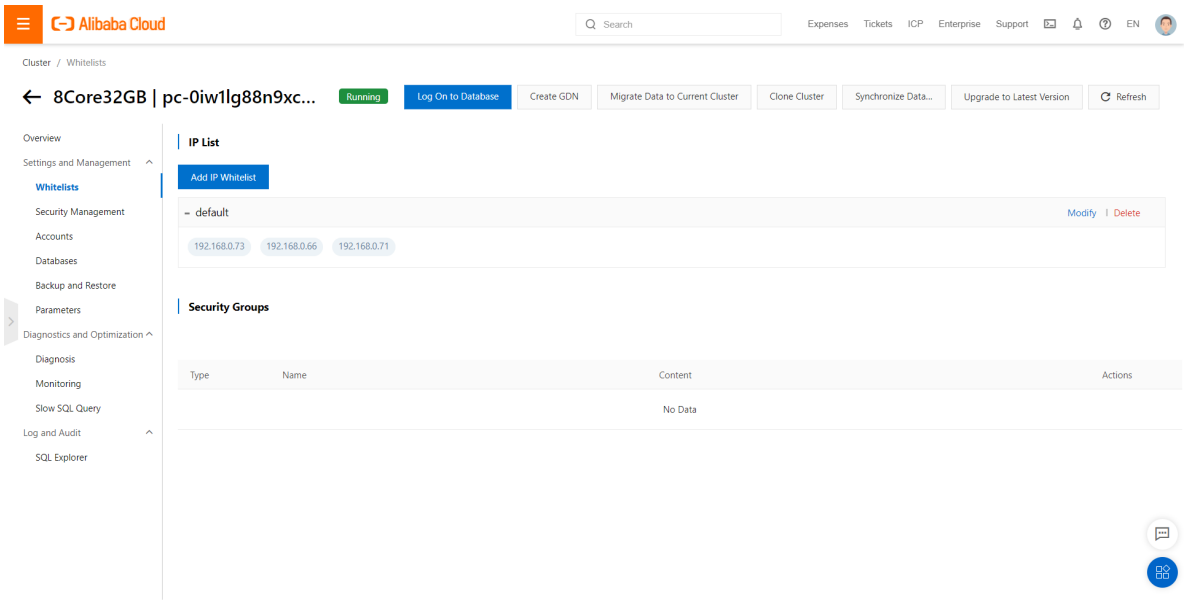
PolarDB-Oracle
Specifications:polar.o.x4.xlarge 8Core32GB
Primary Endpoints:Pgbenchデータ用意するときはこのホストで接続します



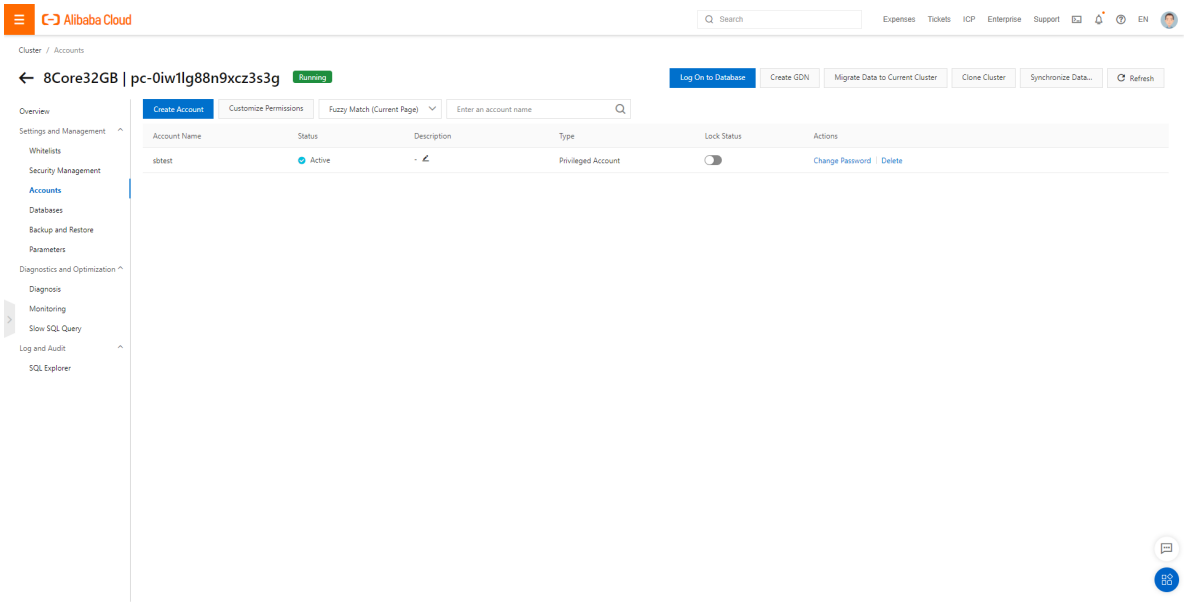
Cluster Endpoints:Pgbenchテストするときはこのホストで接続します



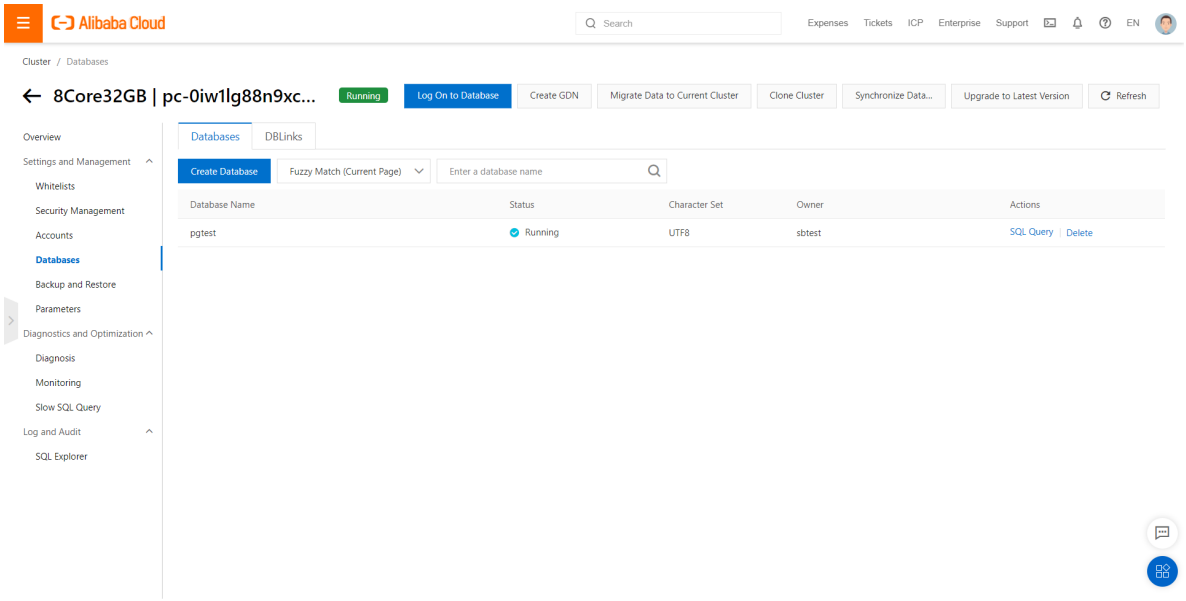
ホワイトリスト：



アカウント：



データベース：



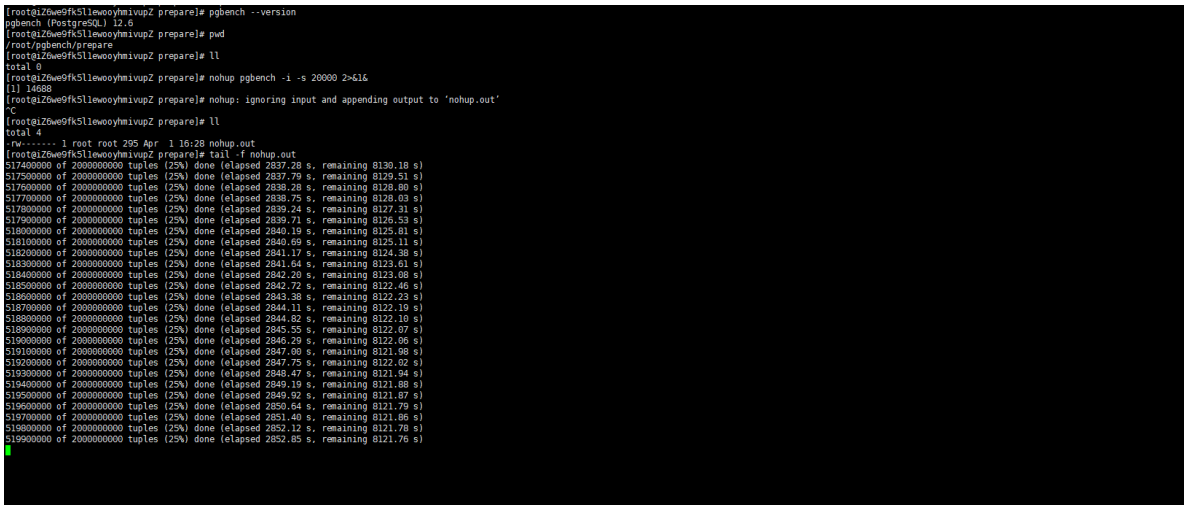
2) pgbenchテストデータを用意する

①Pathを設定：

```
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# export PGHOST=pc-
0iw1lg88n9xcz3s3g.o.polardb.japan.rds.aliyuncs.com
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# export PGPORT=1521
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# export PGDATABASE=pgtest
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# export PGUSER=sbtest
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# export PGPASSWORD=Test1234
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# pgbench --version
pgbench (PostgreSQL) 12.6
```

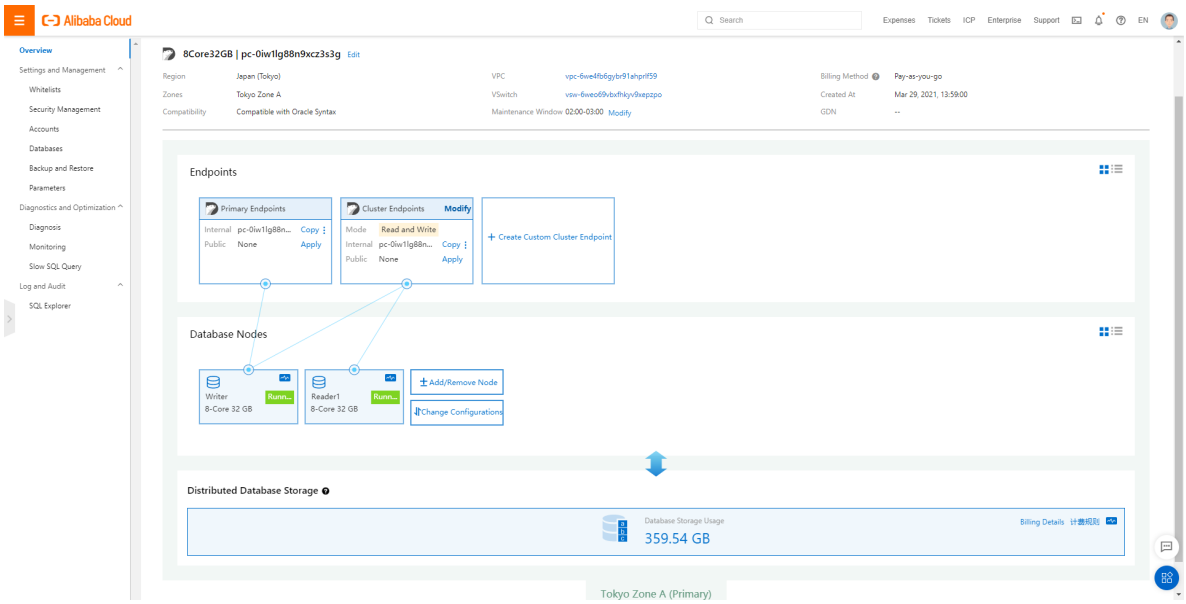
②pgbenchで２０億のデータをデータベースpgtestに初期化する

```
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# pwd
/root/pgbench/prepare
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# ll
total 0
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# nohup pgbench -i -s 20000 2>&1&
[1] 14688
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# nohup: ignoring input and appending
output to ‘nohup.out’
^C
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# ll
total 4
-rw------- 1 root root 295 Apr  1 16:28 nohup.out
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# tail -f nohup.out
517400000 of 2000000000 tuples (25%) done (elapsed 2837.28 s, remaining 8130.18
s)
517500000 of 2000000000 tuples (25%) done (elapsed 2837.79 s, remaining 8129.51
s)
517600000 of 2000000000 tuples (25%) done (elapsed 2838.28 s, remaining 8128.80
s)
517700000 of 2000000000 tuples (25%) done (elapsed 2838.75 s, remaining 8128.03
s)
517800000 of 2000000000 tuples (25%) done (elapsed 2839.24 s, remaining 8127.31
s)
517900000 of 2000000000 tuples (25%) done (elapsed 2839.71 s, remaining 8126.53
s)
```

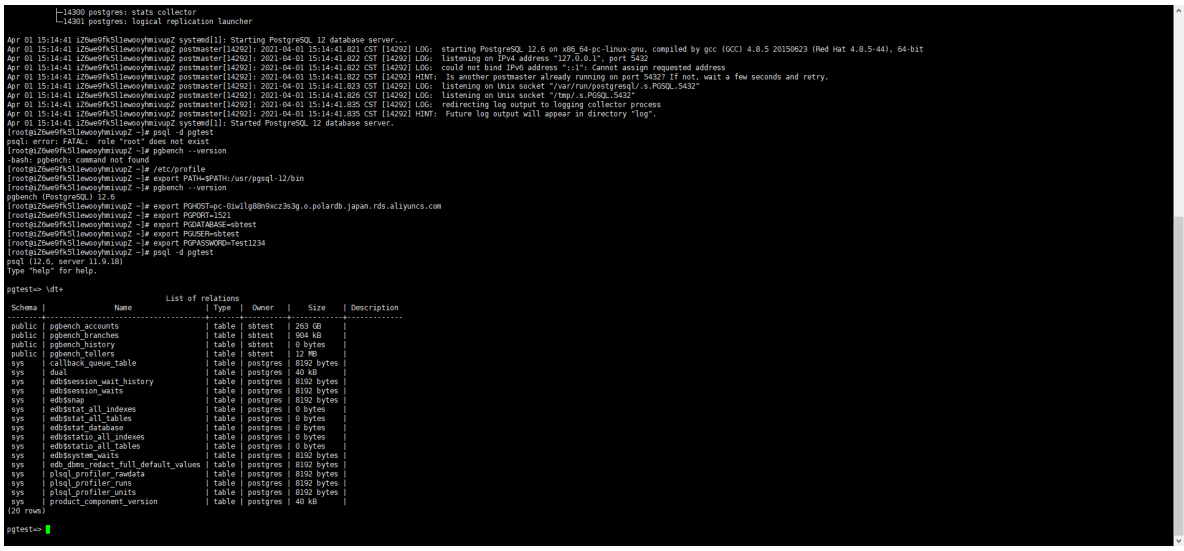


③pgbenchデータの初期化を確認する

```
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# pwd
/root/pgbench/prepare
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# ll
total 1628
-rw------- 1 root root 1666543 Apr  1 22:02 nohup.out
[root@iZ6we9fk5l1ewooyhmivupZ prepare]# tail -f nohup.out
1999400000 of 2000000000 tuples (99%) done (elapsed 11030.34 s, remaining 3.31
s)
1999500000 of 2000000000 tuples (99%) done (elapsed 11030.82 s, remaining 2.76
s)
1999600000 of 2000000000 tuples (99%) done (elapsed 11031.31 s, remaining 2.21
s)
1999700000 of 2000000000 tuples (99%) done (elapsed 11031.80 s, remaining 1.66
s)
1999800000 of 2000000000 tuples (99%) done (elapsed 11032.27 s, remaining 1.10
s)
1999900000 of 2000000000 tuples (99%) done (elapsed 11032.80 s, remaining 0.55
s)
2000000000 of 2000000000 tuples (100%) done (elapsed 11033.28 s, remaining 0.00
s)
vacuuming...
creating primary keys...
done.
```



④データを確認する

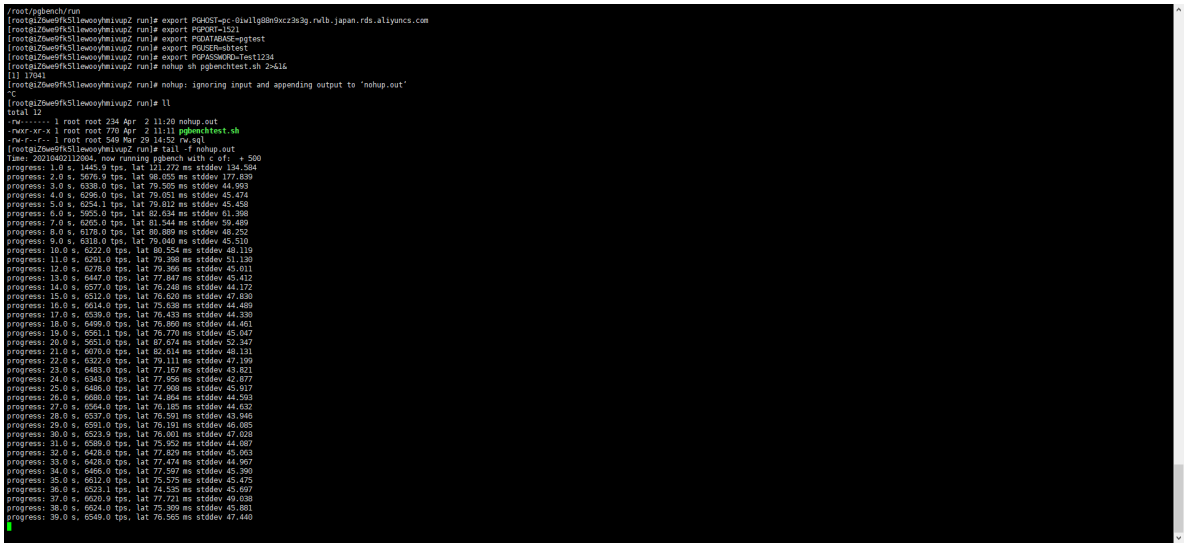


4) pgbenchで性能測定する

①PolarDB-Oracleインスタンスを接続する。
②pgbench性能測定

```
[root@iZ6we9fk5l1ewooyhmivupZ run]# export PGHOST=pc-
0iw1lg88n9xcz3s3g.rdlb.japan.rds.aliyuncs.com
[root@iZ6we9fk5l1ewooyhmivupZ run]# export PGPORT=1521
[root@iZ6we9fk5l1ewooyhmivupZ run]# export PGDATABASE=pgtest
[root@iZ6we9fk5l1ewooyhmivupZ run]# export PGUSER=sbtest
[root@iZ6we9fk5l1ewooyhmivupZ run]# export PGPASSWORD=Test1234
```

```
[root@iZ6we9fk5l1ewooyhmivupZ run]# nohup sh pgbenchtest.sh 2>&1&
```



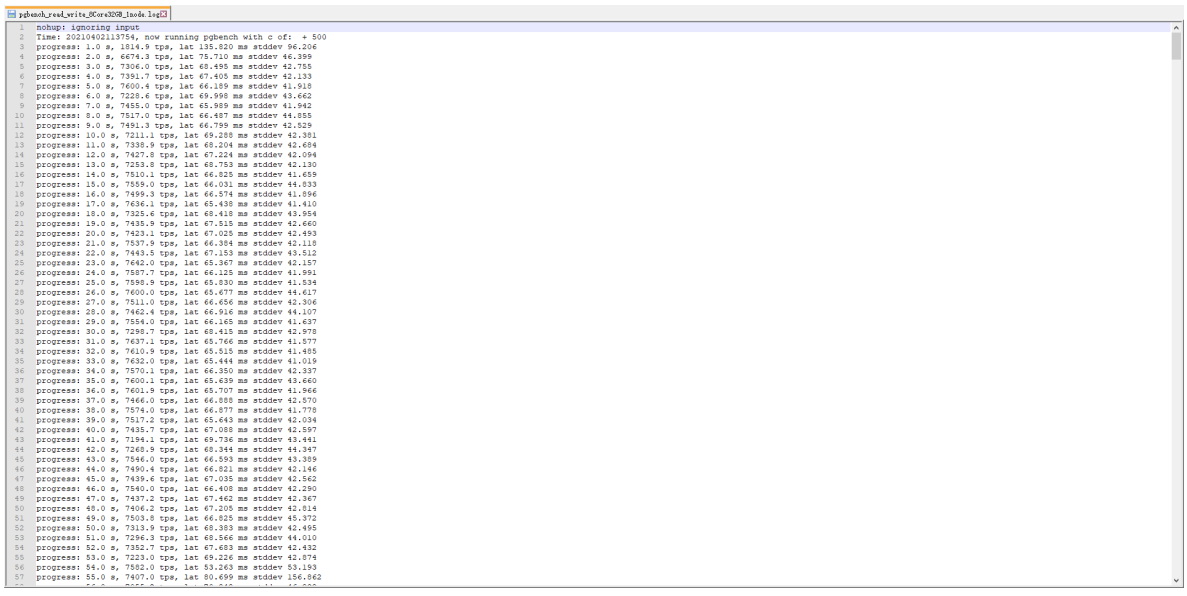
③pgbenchtest.sh

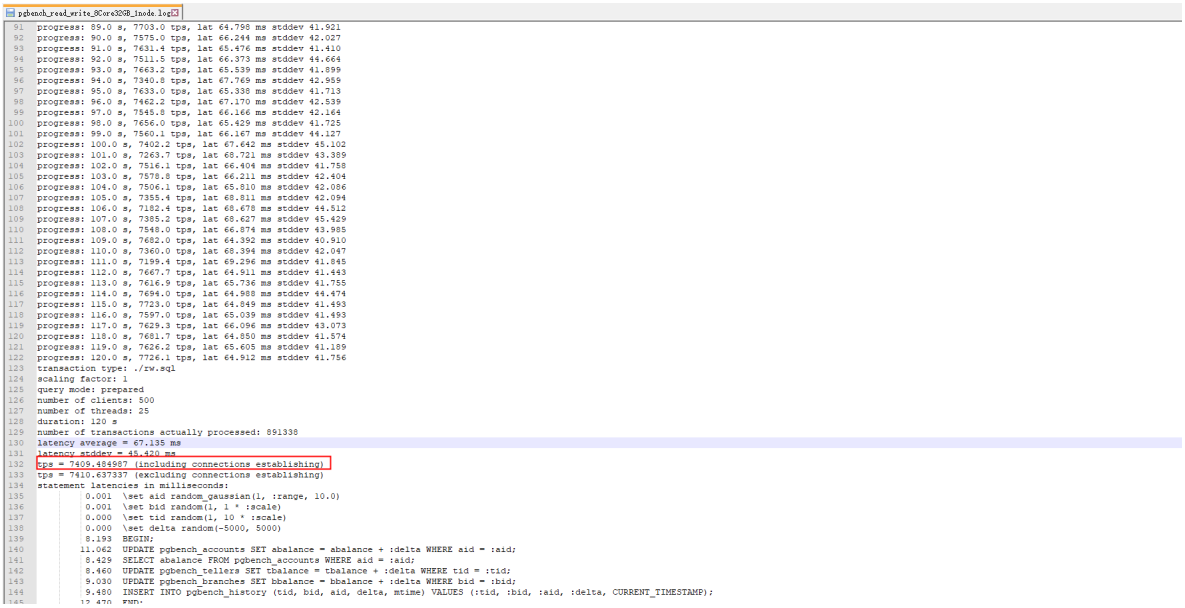
```
#!/bin/sh
#DATE=`date '+%Y%m%d%H%M'`
#mkdir $DATE

# thread=500
# echo "prepare data using default settings, ref sysbench SIZE"  >>
${DATE}/sysbench_read_write_main.log
```

```
# ./sysbench.sh prepare ${thread} >> ${DATE}/sysbench_read_write_main.log

for c in 500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000
do
    #echo "Time: $(date +"%Y%m%d%H%M%S"), now running pgbench with c of: " +
${c}  >> ${DATE}/pgbench_read_write_${c}.log
 # pgbench -M prepared -n -r -P 1 -f ./rw.sql -c ${c} -j 25 -T 120 -D
scale=20000 -D range=2000000000  >> ${DATE}/pgbench_read_write_${c}.log
    echo "Time: $(date +"%Y%m%d%H%M%S"), now running pgbench with c of: " + ${c}

  pgbench -M prepared -n -r -P 1 -f ./rw.sql -c ${c} -j 25 -T 120 -D scale=20000
-D range=2000000000
done
```

rw.sql:

```
\set aid random_gaussian(1, :range, 10.0)
\set bid random(1, 1 * :scale)
\set tid random(1, 10 * :scale)
\set delta random(-5000, 5000)
BEGIN;
UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid,
:aid, :delta, CURRENT_TIMESTAMP);
END;
```





```
pgbench_read_write_8Core32GB_1node.log
progress: 89.0 s, 7703.0 tps, lat 64.798 ms stddev 41.921
progress: 90.0 s, 7575.0 tps, lat 66.244 ms stddev 42.027
progress: 91.0 s, 7631.4 tps, lat 65.476 ms stddev 41.410
progress: 92.0 s, 7511.5 tps, lat 66.373 ms stddev 44.664
progress: 93.0 s, 7663.2 tps, lat 65.539 ms stddev 41.899
progress: 94.0 s, 7340.8 tps, lat 67.769 ms stddev 42.959
progress: 95.0 s, 7633.0 tps, lat 65.338 ms stddev 41.713
progress: 96.0 s, 7462.2 tps, lat 67.170 ms stddev 42.539
progress: 97.0 s, 7545.8 tps, lat 66.166 ms stddev 42.164
progress: 98.0 s, 7656.0 tps, lat 65.429 ms stddev 41.725
progress: 99.0 s, 7560.1 tps, lat 66.167 ms stddev 44.127
progress: 100.0 s, 7402.2 tps, lat 67.642 ms stddev 45.102
progress: 101.0 s, 7263.7 tps, lat 68.721 ms stddev 43.389
progress: 102.0 s, 7516.1 tps, lat 66.404 ms stddev 41.758
progress: 103.0 s, 7570.8 tps, lat 66.211 ms stddev 42.404
progress: 104.0 s, 7506.1 tps, lat 65.810 ms stddev 42.086
progress: 105.0 s, 7355.4 tps, lat 68.811 ms stddev 42.094
progress: 106.0 s, 7182.4 tps, lat 68.678 ms stddev 44.512
progress: 107.0 s, 7385.2 tps, lat 68.627 ms stddev 45.429
progress: 108.0 s, 7548.0 tps, lat 66.874 ms stddev 43.985
progress: 109.0 s, 7682.0 tps, lat 64.392 ms stddev 40.910
progress: 110.0 s, 7360.0 tps, lat 68.394 ms stddev 42.047
progress: 111.0 s, 7199.4 tps, lat 69.296 ms stddev 41.845
progress: 112.0 s, 7667.7 tps, lat 64.911 ms stddev 41.443
progress: 113.0 s, 7616.9 tps, lat 65.736 ms stddev 41.755
progress: 114.0 s, 7694.0 tps, lat 64.988 ms stddev 44.474
progress: 115.0 s, 7723.0 tps, lat 64.849 ms stddev 41.493
progress: 116.0 s, 7597.0 tps, lat 65.039 ms stddev 41.493
progress: 117.0 s, 7629.3 tps, lat 66.096 ms stddev 43.073
progress: 118.0 s, 7681.7 tps, lat 64.850 ms stddev 41.574
progress: 119.0 s, 7626.2 tps, lat 65.605 ms stddev 41.189
progress: 120.0 s, 7726.1 tps, lat 64.912 ms stddev 41.756
transaction type: ./rw.sql
scaling factor: 1
query mode: prepared
number of clients: 500
number of threads: 25
duration: 120 s
number of transactions actually processed: 891338
latency average = 67.135 ms
latency stddev = 45.420 ms
tps = 7409.484987 (including connections establishing)
tps = 7410.637337 (excluding connections establishing)
statement latencies in milliseconds:
         0.001  \set aid random_gaussian(1, :range, 10.0)
         0.001  \set bid random(1, 1 * :scale)
         0.000  \set tid random(1, 10 * :scale)
         0.000  \set delta random(-5000, 5000)
         8.193  BEGIN;
        11.062  UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
         8.429  SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
         8.460  UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
         9.030  UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
         9.480  INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid, :aid, :delta, CURRENT_TIMESTAMP);
        12.470  END;
```

④tpsが実行ログから取得します。QPSではtps*5になる

```
tps = 7409.484987 (including connections establishing)
```

⑤rw.sqlは五つのSQLを実行しているため、qps=tps*5
rw.sql：

```
\set aid random_gaussian(1, :range, 10.0)
\set bid random(1, 1 * :scale)
\set tid random(1, 10 * :scale)
\set delta random(-5000, 5000)
BEGIN;
UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid,
:aid, :delta, CURRENT_TIMESTAMP);
END;
```

ここまでPolarDB-OracleをpgbenchでPerformance測定をご紹介しました