PERCONA

Databases run better with Percona

Creating A Citus Replication Cluster Using Patroni

A HOW TO ...

Robert Bernier

Robert.bernier@percona.com

Senior PostgreSQL Consultant

Percona

Target Environment

PART I: ABOUT ENVIRONMENT

- Introduction:
- Architecture
- State Of Replication
- State Of Patroni Cluster
- postgresql.conf ALA Patroni
- Dynamic Configuration Settings (DCS)

PART II: BUILDING IT

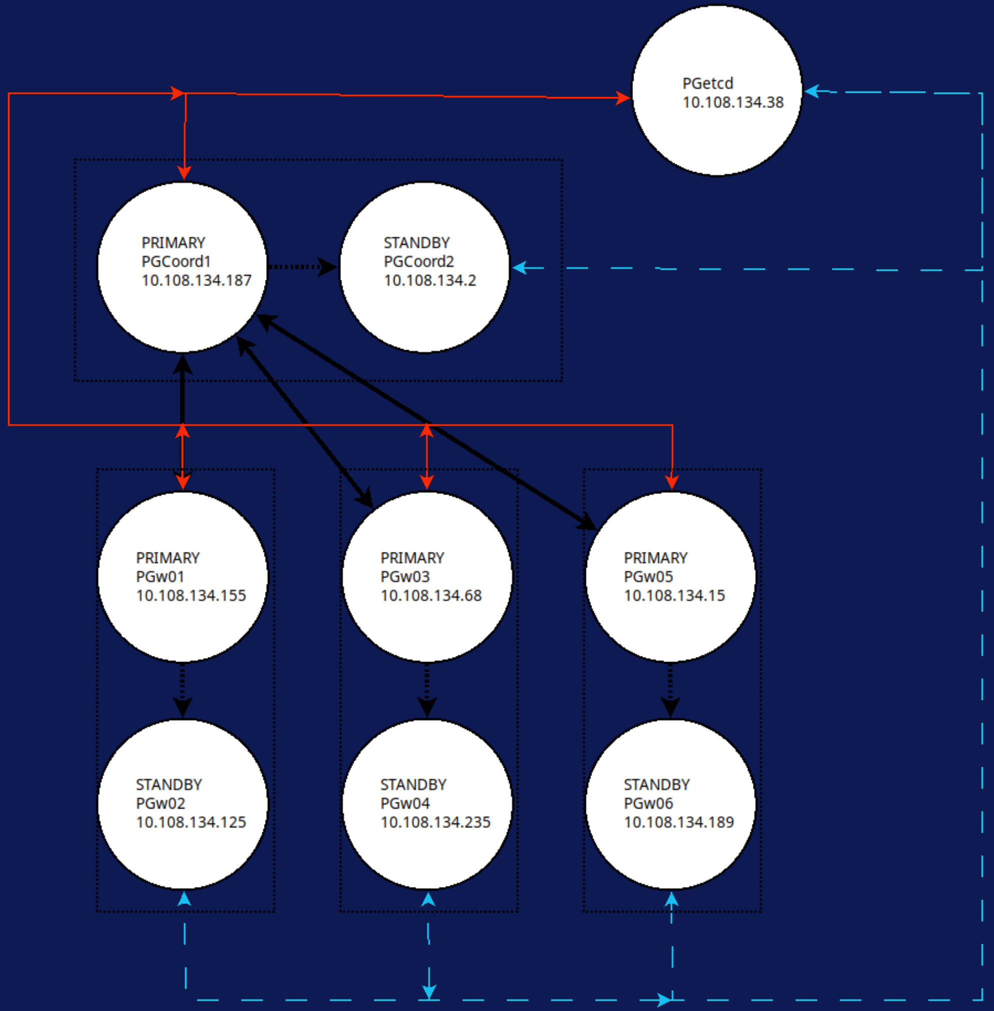
- About Package Versions
- Steps
    - setup the cluster
    - setup etcd
    - create database pgbench
    - update systemd postgres
    - setup patroni
    - start patroni
    - setup pgbench on citus

PART III: WORKING WITH CITUS

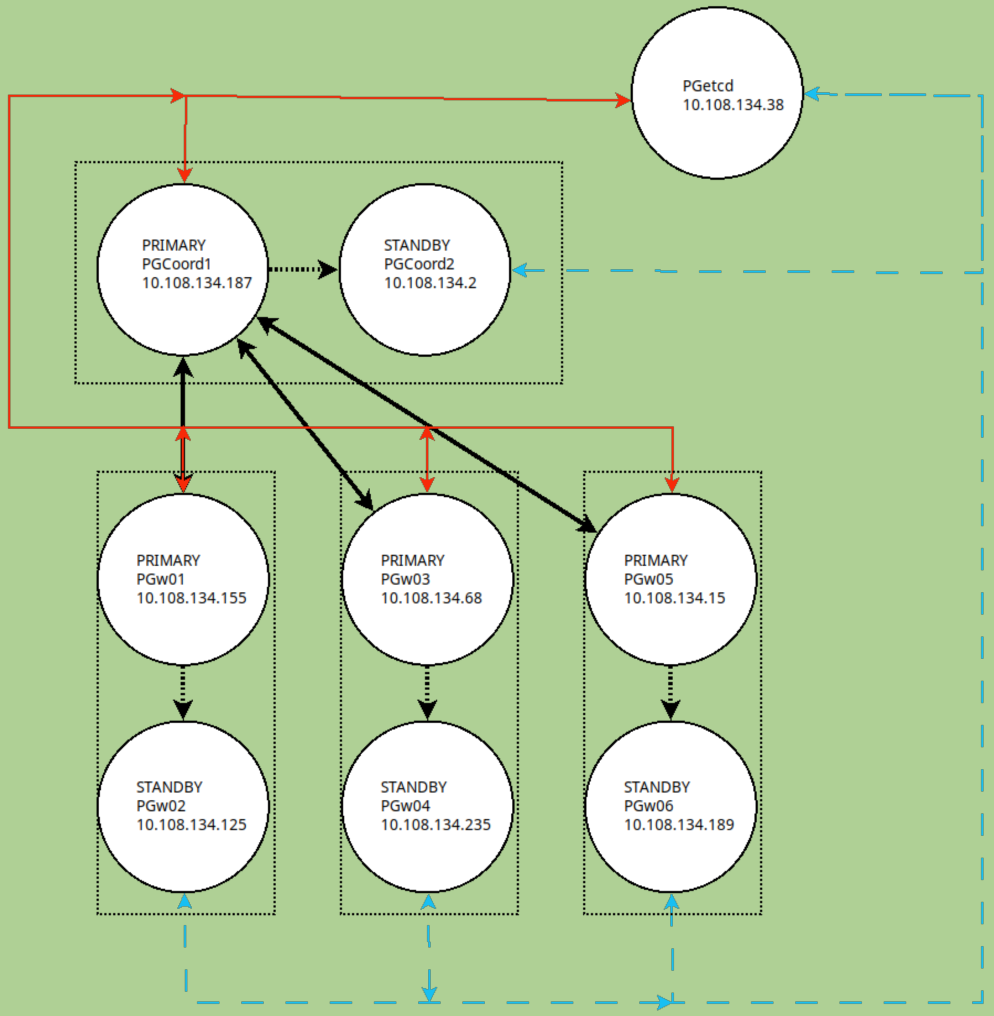
- Citus Concepts
- Configure pgbench Tables

PART I: ABOUT THE ENVIRONMENT

Architecture

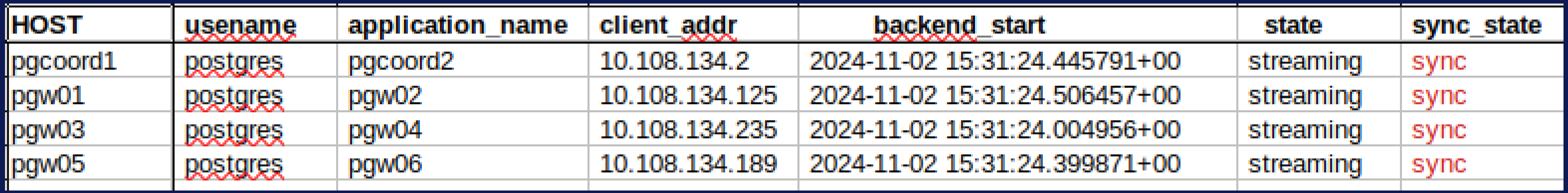


Architecture



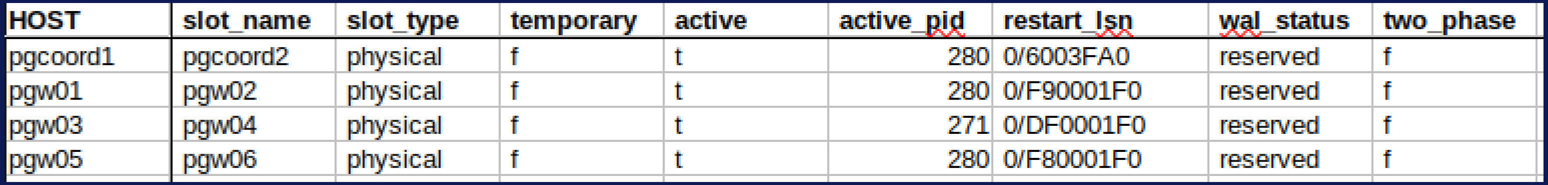
State Of Replication

pg_stat_replication



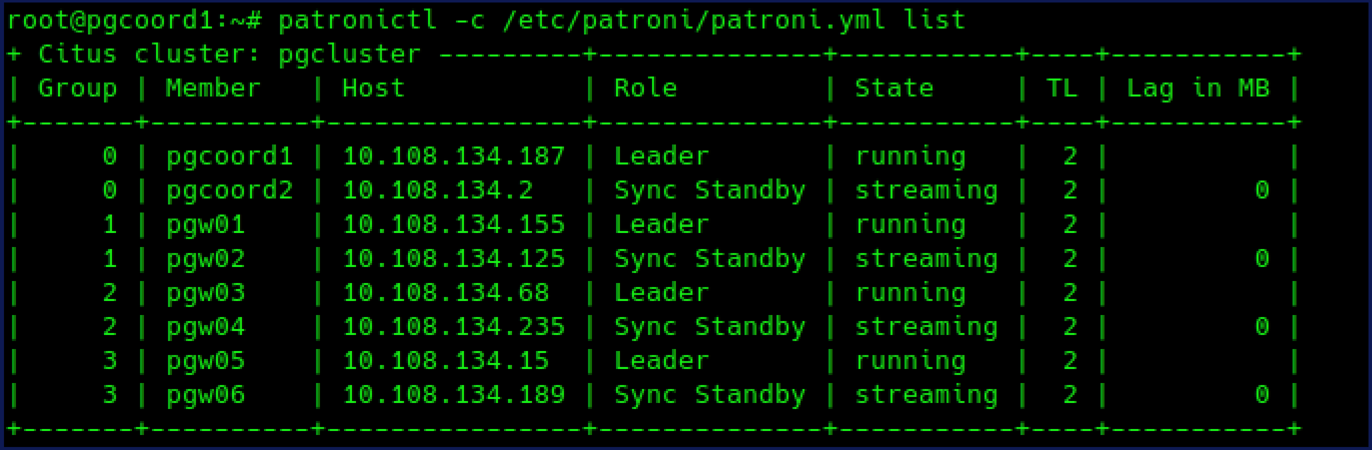
HOST	usename	application_name	client_addr	backend_start	state	sync_state
pgcoord1	postgres	pgcoord2	10.108.134.2	2024-11-02 15:31:24.445791+00	streaming	sync
pgw01	postgres	pgw02	10.108.134.125	2024-11-02 15:31:24.506457+00	streaming	sync
pgw03	postgres	pgw04	10.108.134.235	2024-11-02 15:31:24.004956+00	streaming	sync
pgw05	postgres	pgw06	10.108.134.189	2024-11-02 15:31:24.399871+00	streaming	sync

pg_get_replication_slots



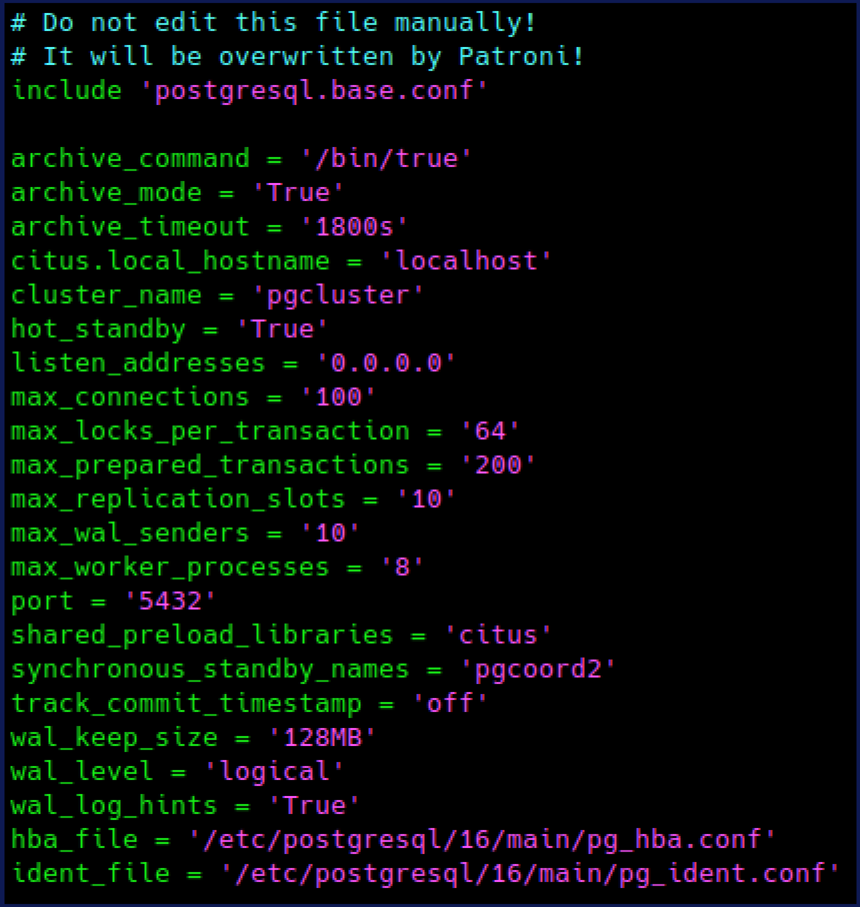
HOST	slot_name	slot_type	temporary	active	active_pid	restart_lsn	wal_status	two_phase
pgcoord1	pgcoord2	physical	f	t	280	0/6003FA0	reserved	f
pgw01	pgw02	physical	f	t	280	0/F90001F0	reserved	f
pgw03	pgw04	physical	f	t	271	0/DF0001F0	reserved	f
pgw05	pgw06	physical	f	t	280	0/F80001F0	reserved	f

State Of Patroni Cluster



```
root@pgcoord1:~# patronictl -c /etc/patroni/patroni.yml list
+ Citus cluster: pgcluster --------+--------------+-----------+----+-----------+
| Group | Member   | Host           | Role         | State     | TL | Lag in MB |
+-------+----------+----------------+--------------+-----------+----+-----------+
|     0 | pgcoord1 | 10.108.134.187 | Leader       | running   |  2 |           |
|     0 | pgcoord2 | 10.108.134.2   | Sync Standby | streaming |  2 |         0 |
|     1 | pgw01    | 10.108.134.155 | Leader       | running   |  2 |           |
|     1 | pgw02    | 10.108.134.125 | Sync Standby | streaming |  2 |         0 |
|     2 | pgw03    | 10.108.134.68  | Leader       | running   |  2 |           |
|     2 | pgw04    | 10.108.134.235 | Sync Standby | streaming |  2 |         0 |
|     3 | pgw05    | 10.108.134.15  | Leader       | running   |  2 |           |
|     3 | pgw06    | 10.108.134.189 | Sync Standby | streaming |  2 |         0 |
+-------+----------+----------------+--------------+-----------+----+-----------+
```

postgresql.conf ALA Patroni



```
# Do not edit this file manually!
# It will be overwritten by Patroni!
include 'postgresql.base.conf'

archive_command = '/bin/true'
archive_mode = 'True'
archive_timeout = '1800s'
citus.local_hostname = 'localhost'
cluster_name = 'pgcluster'
hot_standby = 'True'
listen_addresses = '0.0.0.0'
max_connections = '100'
max_locks_per_transaction = '64'
max_prepared_transactions = '200'
max_replication_slots = '10'
max_wal_senders = '10'
max_worker_processes = '8'
port = '5432'
shared_preload_libraries = 'citus'
synchronous_standby_names = 'pgcoord2'
track_commit_timestamp = 'off'
wal_keep_size = '128MB'
wal_level = 'logical'
wal_log_hints = 'True'
hba_file = '/etc/postgresql/16/main/pg_hba.conf'
ident_file = '/etc/postgresql/16/main/pg_ident.conf'
```

PART II: BUILDING THE ENVIRONMENT

Version HELL

```
Percona Distribution is used:
    PostgreSQL
    etcd
    patroni
```

```
  Ubuntu version: 24.04     (noble)
postgres version: 16
 patroni version: 3.3.2     (minimum required 3.*
                             ATTENTION community version: 3.2.2
                                          uptodate version: 4.0.3)
    etcd version: 3.5.15    (ATTENTION community version: 3.4.30)
   citus version: 12.1
```

REFERENCES

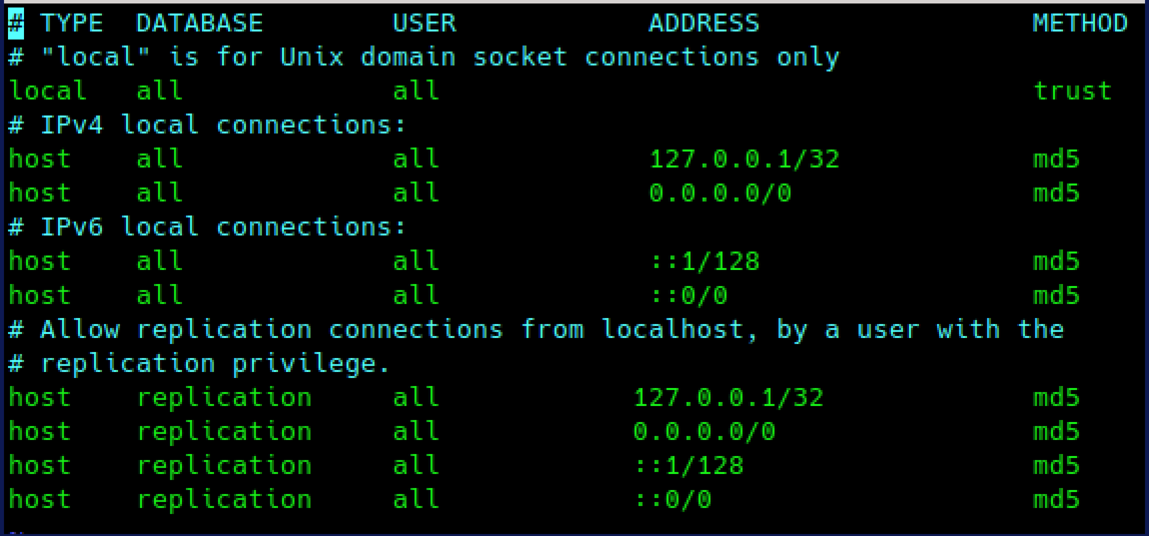
https://www.percona.com/postgresql/software/postgresql-distribution
https://patroni.readthedocs.io/en/rel_3_3/
https://patroni.readthedocs.io/en/latest/index.html
https://etcd.io/docs/
https://www.citusdata.com/download/
https://github.com/rbernierZulu/pg_conf_Seattle-2024

Building The Cluster

- Steps
    - install software packages
    - initialize, configure the cluster
    - setup etcd
    - create database pgbench
    - update systemd postgres
    - setup patroni
    - start patroni
    - setup pgbench on citus

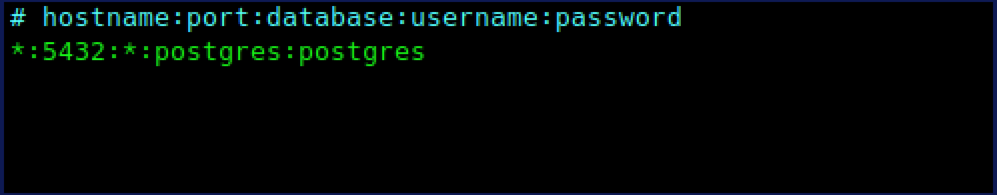
Setup The PG Hosts

pg_hba.conf



```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
host    all             all             0.0.0.0/0               md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
host    all             all             ::0/0                   md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
host    replication     all            127.0.0.1/32             md5
host    replication     all            0.0.0.0/0                md5
host    replication     all            ::1/128                  md5
host    replication     all            ::0/0                    md5
```

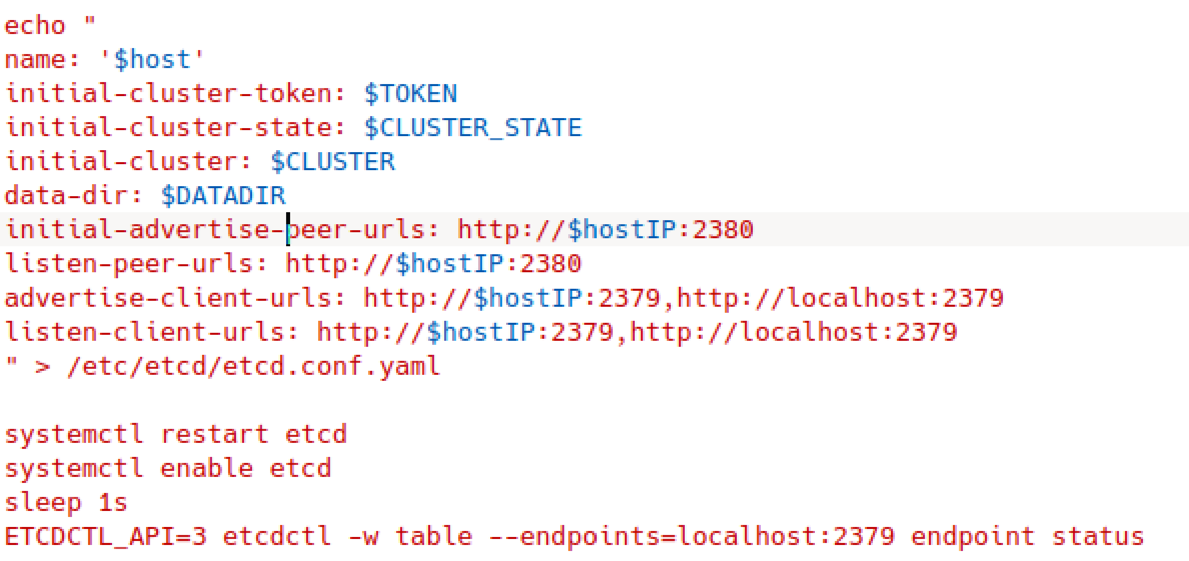
$HOME/.pgpass



```
# hostname:port:database:username:password
*:5432:*:postgres:postgres
```

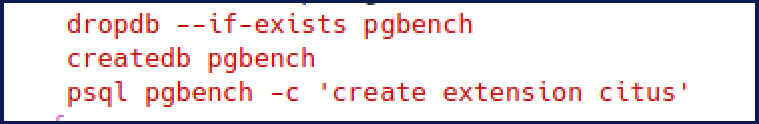
Setup ETCD

/etc/etcd/etcd.conf.yaml



```
echo "
name: '$host'
initial-cluster-token: $TOKEN
initial-cluster-state: $CLUSTER_STATE
initial-cluster: $CLUSTER
data-dir: $DATADIR
initial-advertise-peer-urls: http://$hostIP:2380
listen-peer-urls: http://$hostIP:2380
advertise-client-urls: http://$hostIP:2379,http://localhost:2379
listen-client-urls: http://$hostIP:2379,http://localhost:2379
" > /etc/etcd/etcd.conf.yaml

systemctl restart etcd
systemctl enable etcd
sleep 1s
ETCDCTL_API=3 etcdctl -w table --endpoints=localhost:2379 endpoint status
```

Create Database pgbench



```
dropdb --if-exists pgbench
createdb pgbench
psql pgbench -c 'create extension citus'
```

Caveat: `The database cannot be dropped once citus is active`

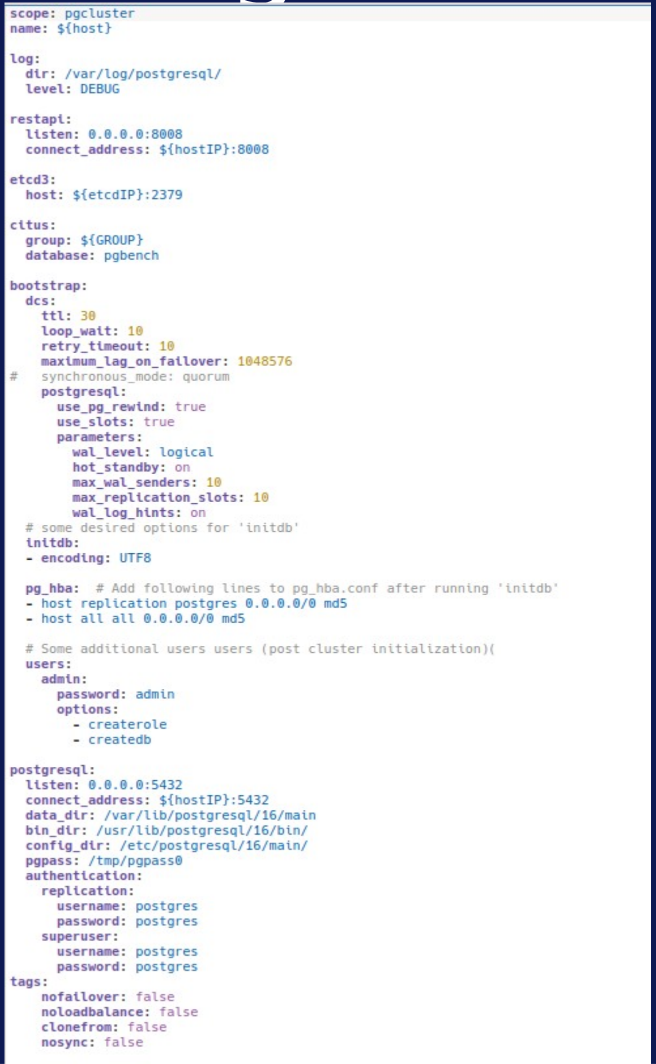
Update systemd

```
# on all hosts
systemctl disable postgresql@16-main
systemctl stop postgresql@16-main
```

```
# on all STANDBY hosts
rm -rf /var/lib/postgresql/16/main/*
```

Configure Patroni

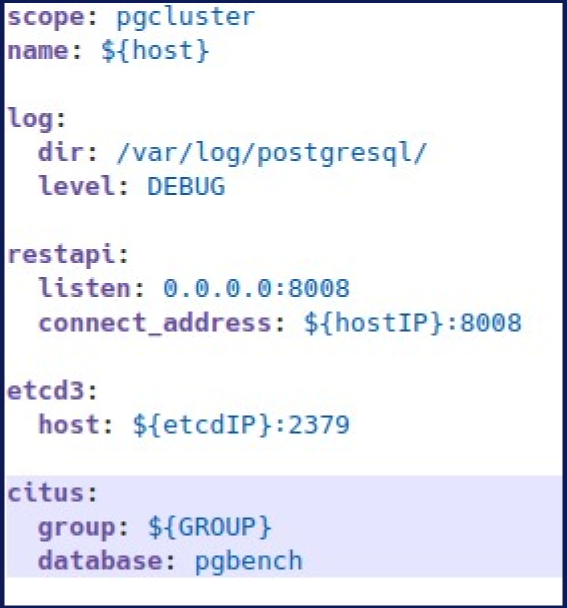
```
vim /etc/patroni/patroni.yml
```



```yaml
scope: pgcluster
name: ${host}

log:
  dir: /var/log/postgresql/
  level: DEBUG

restapi:
  listen: 0.0.0.0:8008
  connect_address: ${hostIP}:8008

etcd3:
  host: ${etcdIP}:2379

citus:
  group: ${GROUP}
  database: pgbench

bootstrap:
  dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 1048576
#   synchronous_mode: quorum
    postgresql:
      use_pg_rewind: true
      use_slots: true
      parameters:
        wal_level: logical
        hot_standby: on
        max_wal_senders: 10
        max_replication_slots: 10
        wal_log_hints: on
  # some desired options for 'initdb'
  initdb:
  - encoding: UTF8

  pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
  - host replication postgres 0.0.0.0/0 md5
  - host all all 0.0.0.0/0 md5

  # Some additional users users (post cluster initialization)(
  users:
    admin:
      password: admin
      options:
        - createrole
        - createdb

postgresql:
  listen: 0.0.0.0:5432
  connect_address: ${hostIP}:5432
  data_dir: /var/lib/postgresql/16/main
  bin_dir: /usr/lib/postgresql/16/bin/
  config_dir: /etc/postgresql/16/main/
  pgpass: /tmp/pgpass0
  authentication:
    replication:
      username: postgres
      password: postgres
    superuser:
      username: postgres
      password: postgres
tags:
    nofailover: false
    noloadbalance: false
    clonefrom: false
    nosync: false
```

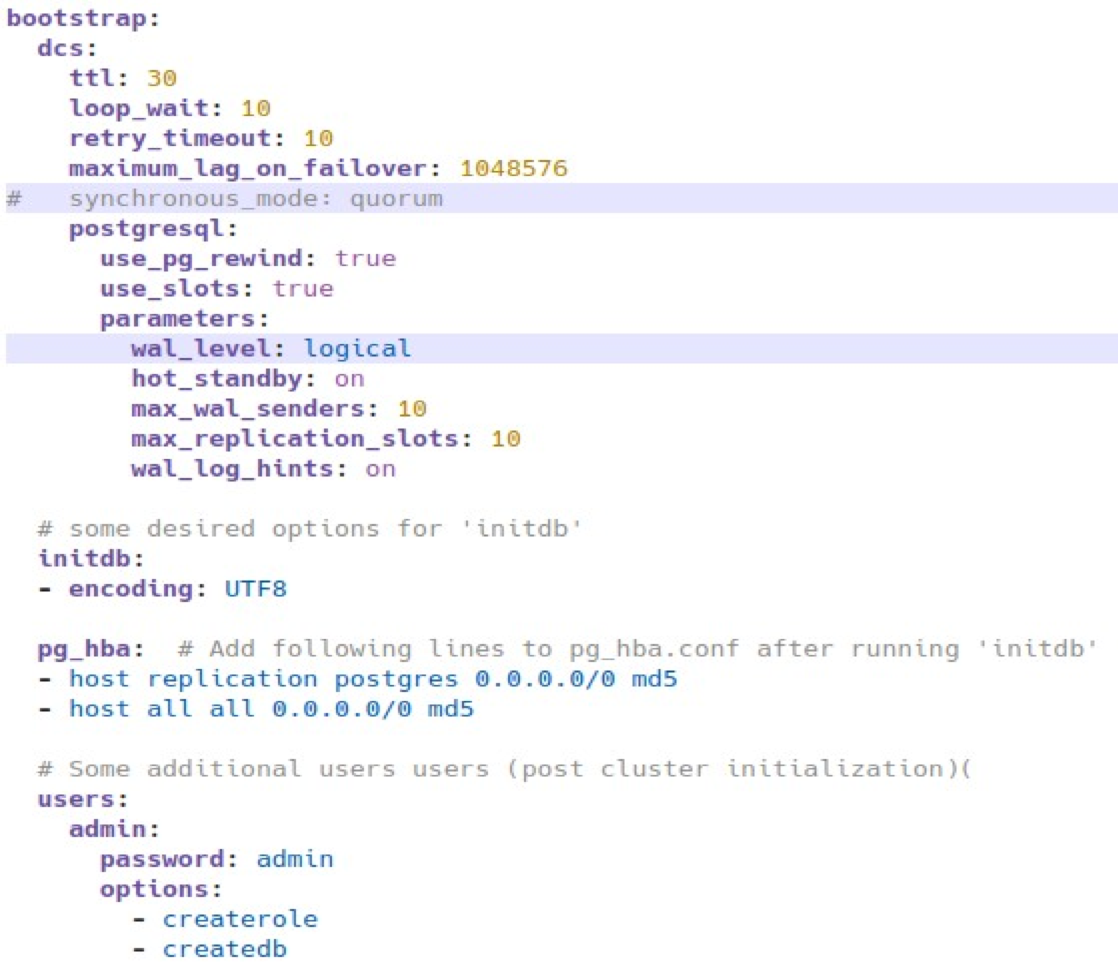
Configure Patroni Cont’d

```
vim /etc/patroni/patroni.yml
```



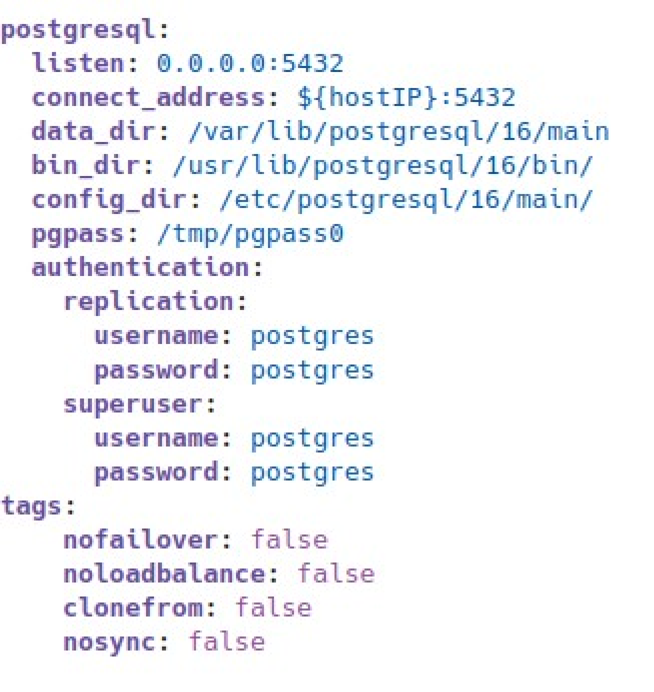
```
scope: pgcluster
name: ${host}

log:
  dir: /var/log/postgresql/
  level: DEBUG

restapi:
  listen: 0.0.0.0:8008
  connect_address: ${hostIP}:8008

etcd3:
  host: ${etcdIP}:2379

citus:
  group: ${GROUP}
  database: pgbench
```

Configure Patroni Cont’d



```yaml
bootstrap:
  dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 1048576
#   synchronous_mode: quorum
    postgresql:
      use_pg_rewind: true
      use_slots: true
      parameters:
        wal_level: logical
        hot_standby: on
        max_wal_senders: 10
        max_replication_slots: 10
        wal_log_hints: on

  # some desired options for 'initdb'
  initdb:
  - encoding: UTF8

  pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
  - host replication postgres 0.0.0.0/0 md5
  - host all all 0.0.0.0/0 md5

  # Some additional users users (post cluster initialization)(
  users:
    admin:
      password: admin
      options:
        - createrole
        - createdb
```

Configure Patroni Cont’d

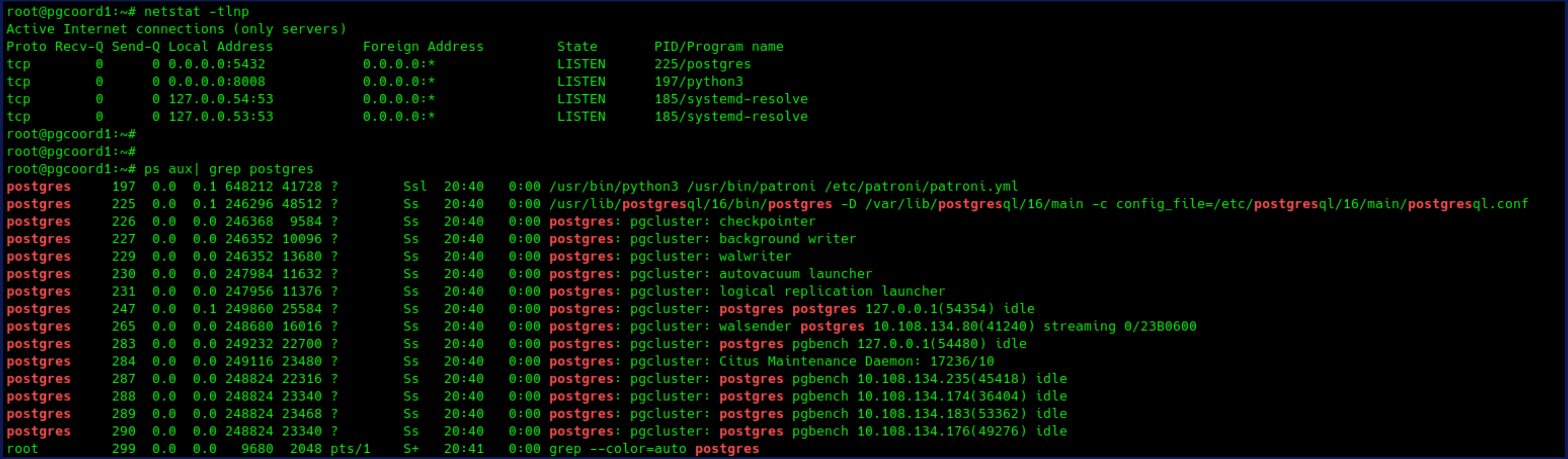


```
postgresql:
  listen: 0.0.0.0:5432
  connect_address: ${hostIP}:5432
  data_dir: /var/lib/postgresql/16/main
  bin_dir: /usr/lib/postgresql/16/bin/
  config_dir: /etc/postgresql/16/main/
  pgpass: /tmp/pgpass0
  authentication:
    replication:
      username: postgres
      password: postgres
    superuser:
      username: postgres
      password: postgres
tags:
    nofailover: false
    noloadbalance: false
    clonefrom: false
    nosync: false
```

Start Patroni

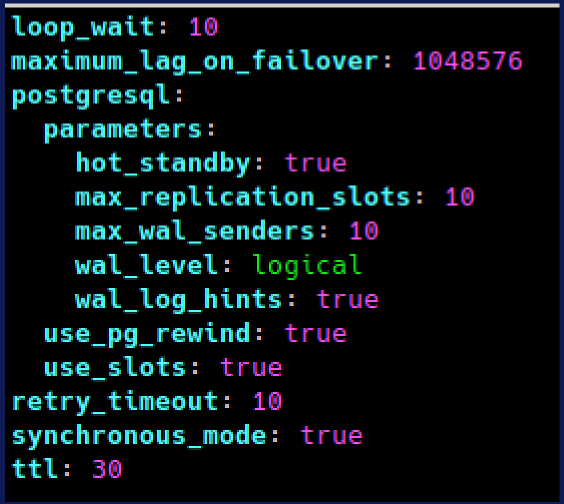
```
# Validate:
patroni --validate-config /etc/patroni/patroni


# Test:
patroni /etc/patroni/patroni


# Start Service:
systemctl start patroni
```

Start Patroni Cont’d



```
root@pgcoord1:~# netstat -tlnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN      225/postgres
tcp        0      0 0.0.0.0:8008            0.0.0.0:*               LISTEN      197/python3
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      185/systemd-resolve
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      185/systemd-resolve
root@pgcoord1:~#
root@pgcoord1:~#
root@pgcoord1:~# ps aux| grep postgres
postgres     197  0.0  0.1 648212 41728 ?        Ssl  20:40   0:00 /usr/bin/python3 /usr/bin/patroni /etc/patroni/patroni.yml
postgres     225  0.0  0.1 246296 48512 ?        Ss   20:40   0:00 /usr/lib/postgresql/16/bin/postgres -D /var/lib/postgresql/16/main -c config_file=/etc/postgresql/16/main/postgresql.conf
postgres     226  0.0  0.0 246368  9584 ?        Ss   20:40   0:00 postgres: pgcluster: checkpointer
postgres     227  0.0  0.0 246352 10096 ?        Ss   20:40   0:00 postgres: pgcluster: background writer
postgres     229  0.0  0.0 246352 13680 ?        Ss   20:40   0:00 postgres: pgcluster: walwriter
postgres     230  0.0  0.0 247984 11632 ?        Ss   20:40   0:00 postgres: pgcluster: autovacuum launcher
postgres     231  0.0  0.0 247956 11376 ?        Ss   20:40   0:00 postgres: pgcluster: logical replication launcher
postgres     247  0.0  0.1 249860 25584 ?        Ss   20:40   0:00 postgres: pgcluster: postgres postgres 127.0.0.1(54354) idle
postgres     265  0.0  0.0 248680 16016 ?        Ss   20:40   0:00 postgres: pgcluster: walsender postgres 10.108.134.80(41240) streaming 0/23B0600
postgres     283  0.0  0.0 249232 22700 ?        Ss   20:40   0:00 postgres: pgcluster: postgres pgbench 127.0.0.1(54480) idle
postgres     284  0.0  0.0 249116 23480 ?        Ss   20:40   0:00 postgres: pgcluster: Citus Maintenance Daemon: 17236/10
postgres     287  0.0  0.0 248824 22316 ?        Ss   20:40   0:00 postgres: pgcluster: postgres pgbench 10.108.134.235(45418) idle
postgres     288  0.0  0.0 248824 23340 ?        Ss   20:40   0:00 postgres: pgcluster: postgres pgbench 10.108.134.174(36404) idle
postgres     289  0.0  0.0 248824 23468 ?        Ss   20:40   0:00 postgres: pgcluster: postgres pgbench 10.108.134.183(53362) idle
postgres     290  0.0  0.0 248824 23340 ?        Ss   20:40   0:00 postgres: pgcluster: postgres pgbench 10.108.134.176(49276) idle
root         299  0.0  0.0   9680  2048 pts/1    S+   20:41   0:00 grep --color=auto postgres
```

Dynamic Configuration Settings (DCS)



```
loop_wait: 10
maximum_lag_on_failover: 1048576
postgresql:
  parameters:
    hot_standby: true
    max_replication_slots: 10
    max_wal_senders: 10
    wal_level: logical
    wal_log_hints: true
  use_pg_rewind: true
  use_slots: true
retry_timeout: 10
synchronous_mode: true
ttl: 30
```

Startup Order

```
PRIMARY
    pgcoord1
    pgw01
    pgw03
    pgw05

STANDBY
    pgcoord2
    pgw02
    pgw04
    pgw06
```

PART III: WORKING WITH CITUS

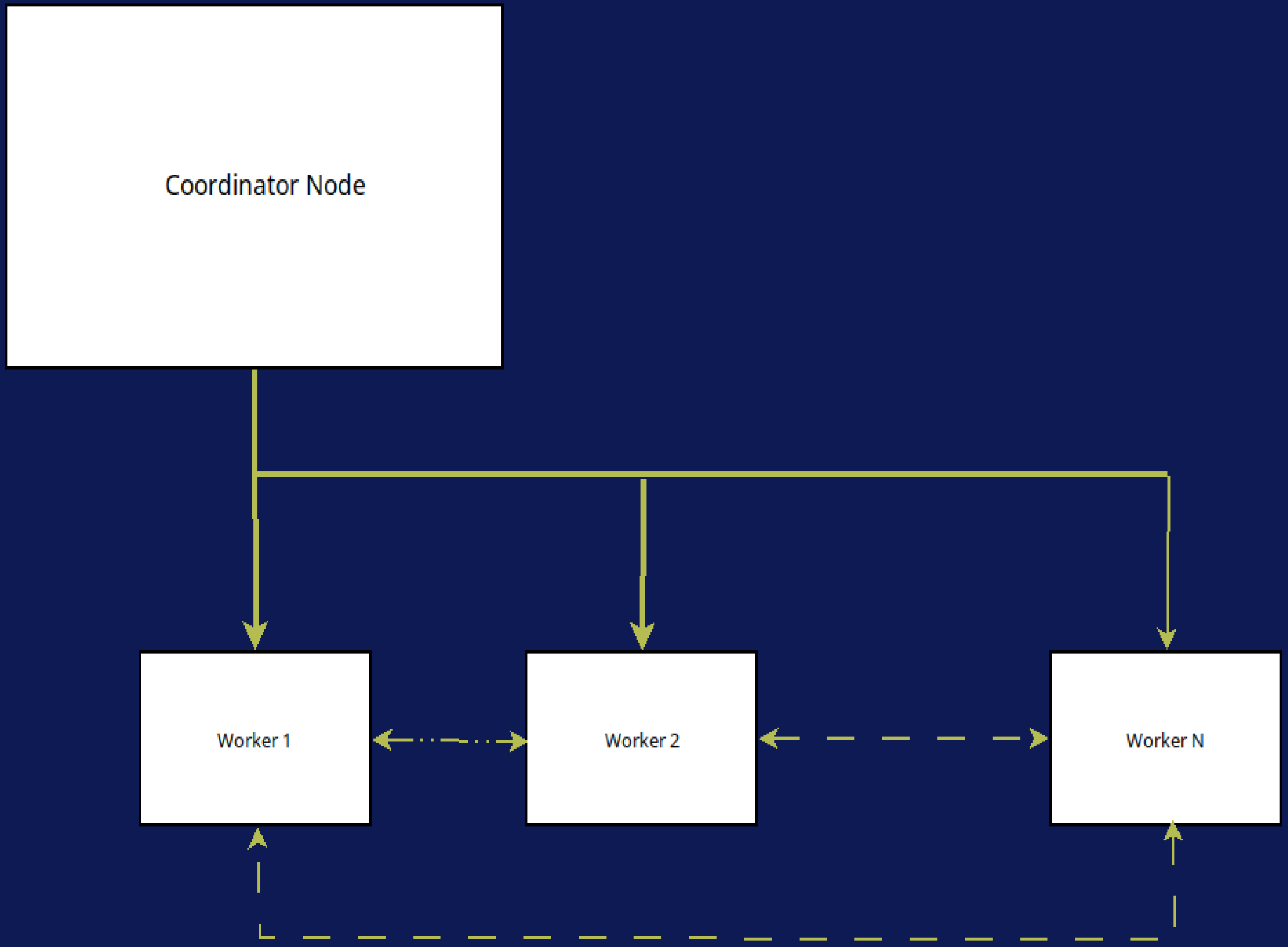
What is CitusDB

An extension that:

- horizontally scales PostgreSQL
- uses sharding and replication.
- Parallelizes SQL queries
- Can create column wise tables

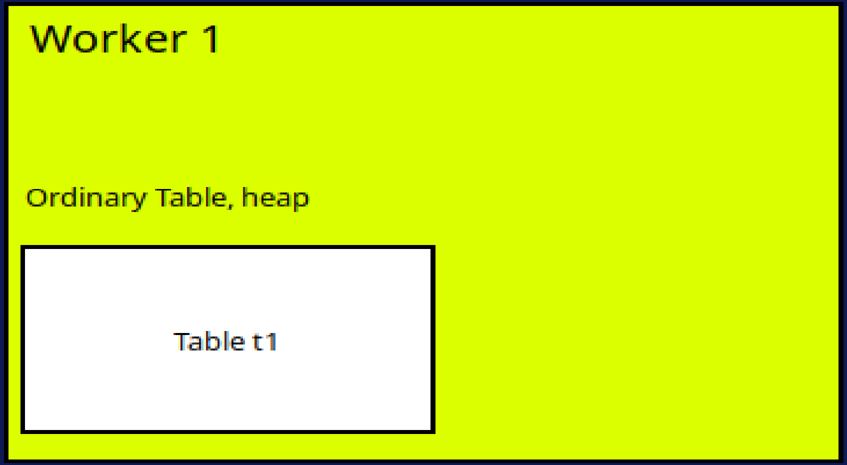
Some Citus Concepts

```
NODES

- coordinator
- worker
```



Some Citus Concepts

`Ordinary Table (heap)`



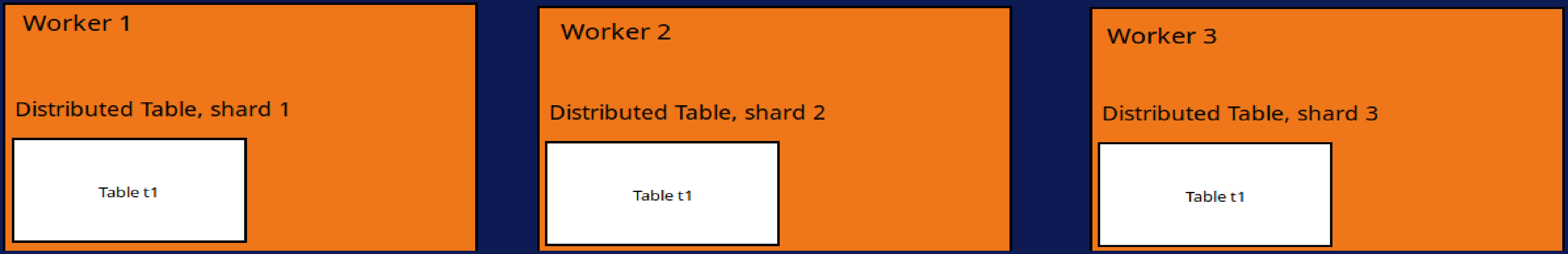
Worker 1

Ordinary Table, heap

Table t1

Some Citus Concepts

`Distributed Table`



Worker 1

Distributed Table, shard 1

Table t1

Worker 2

Distributed Table, shard 2

Table t1

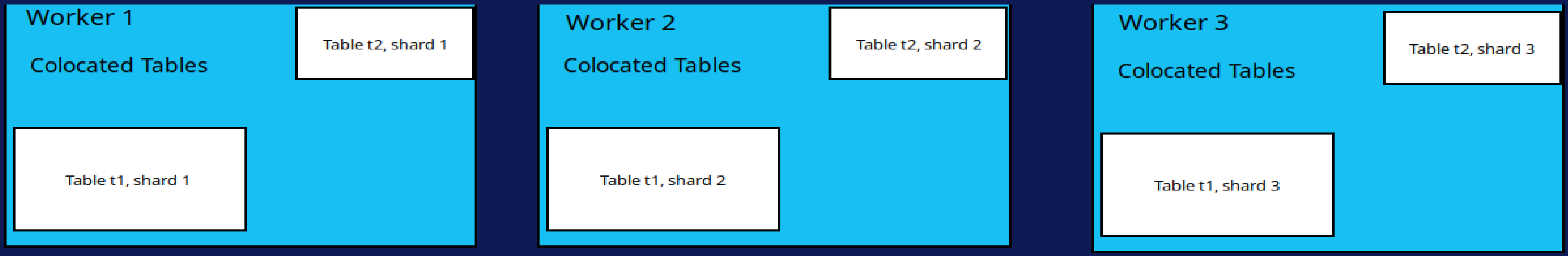
Worker 3

Distributed Table, shard 3

Table t1

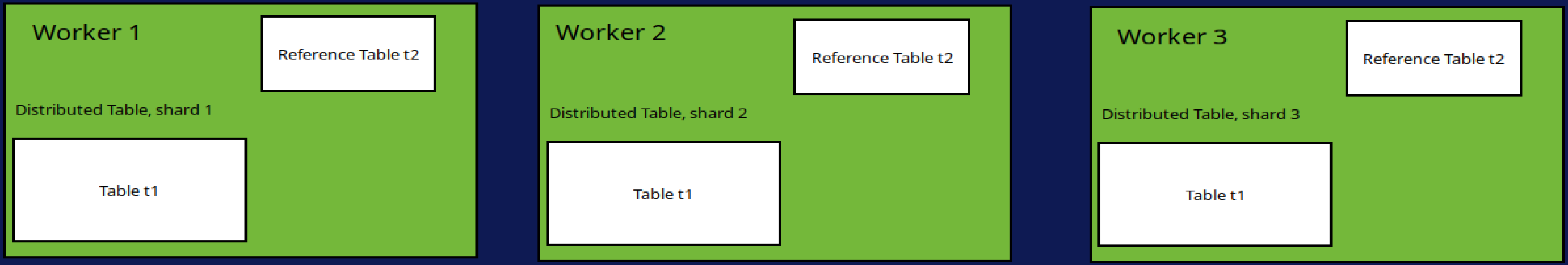
Some Citus Concepts

Distributed table, Colocated (foreign keys)



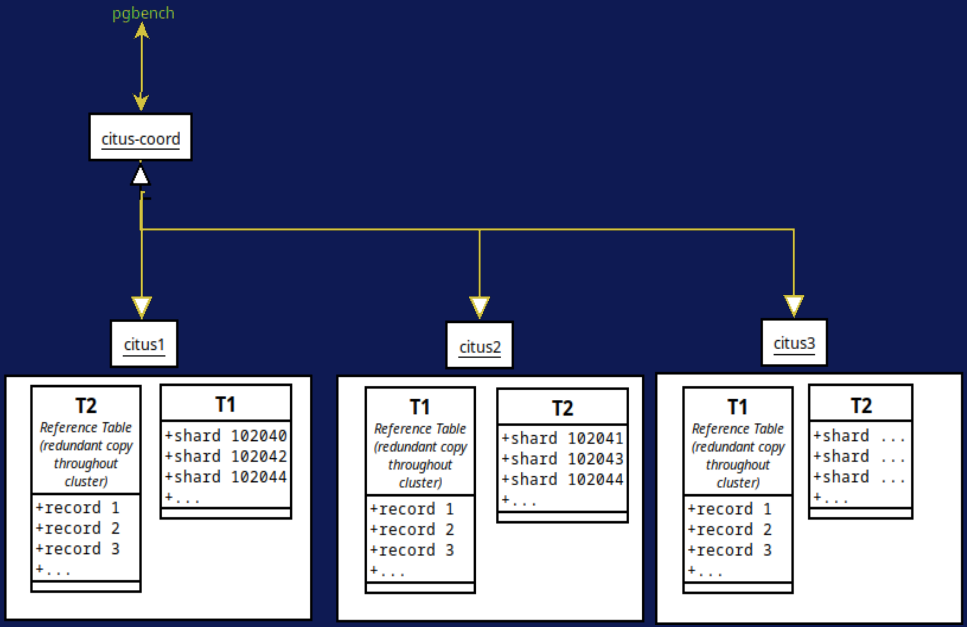
Some Citus Concepts

`Reference Table (data redundancy)`



CitusDB Horizontal Scaling

Three Worker Node, reference tables



```
Execute On Coordinator Node

-- drop tables
drop table if exists pgbench_accounts,pgbench_branches,
                     pgbench_history,pgbench_tellers;

---------------------------------
-- ADD node
select citus_add_node('citus3', 5432);
```

```
Create Reference & Distributed Tables Across Cluster

-- create new tables
create table t1(id serial primary key, comment text);

create table t2 (
    id serial primary key,
    t1_fk int references t1(id) default (random()*1000)::int%4
);

------------------------------------------
select create_reference_table('t1');
select create_distributed_table('t2', 'id');
```

```
Data Population


insert into t1
values (0,'apple')
      ,(1,'oranges')
      ,(2,'pineapple')
      ,(3,'strawberry');

insert into t2 (select * from generate_series(1,1E6)t1);
```

Configure pgbench Tables

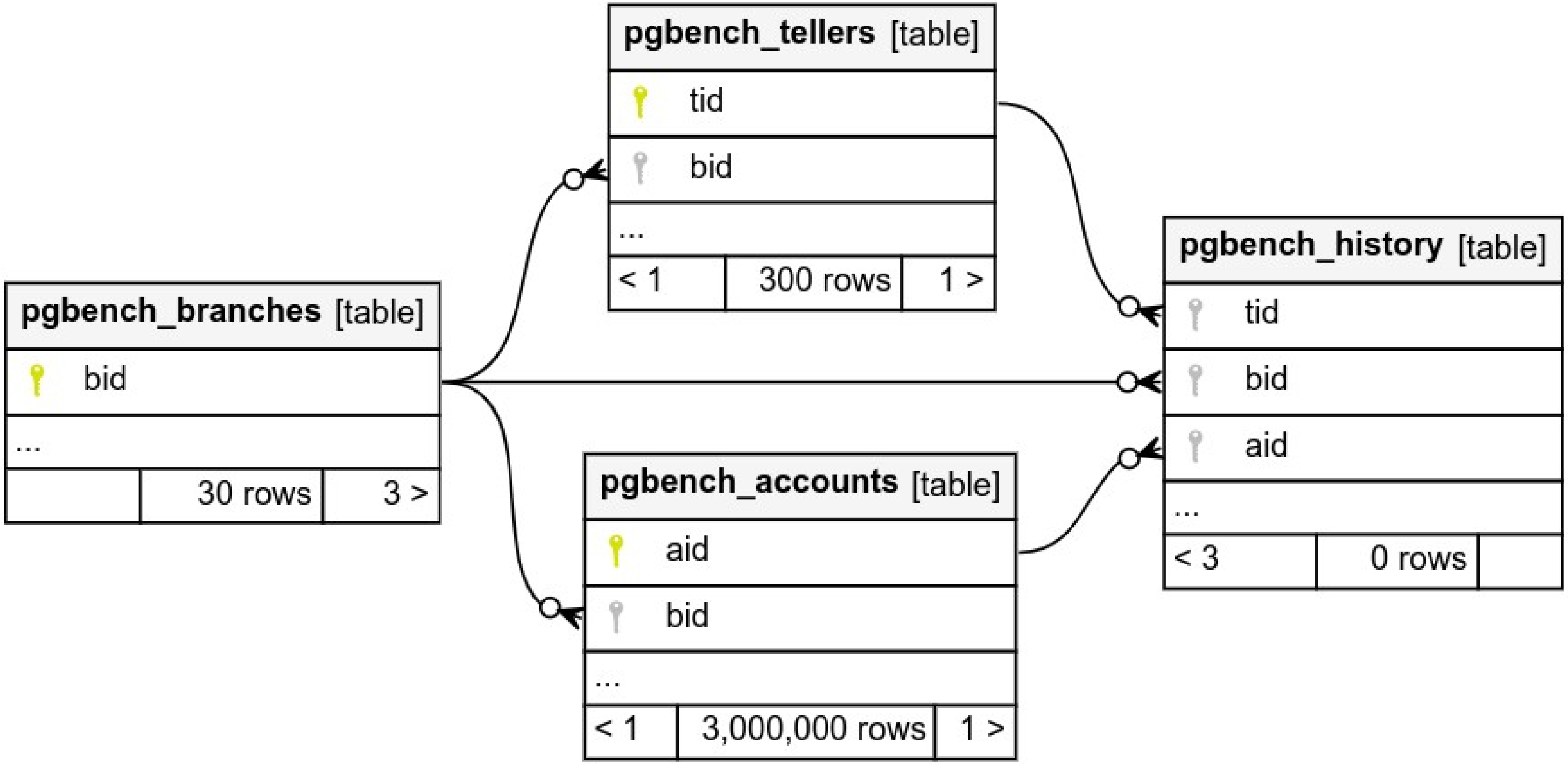
```
# on the command line
pgbench  -iI t pgbench
```

```
-- pgcoord1
alter table pgbench_history replica identity full;

select create_distributed_table('pgbench_accounts', 'aid');
select create_reference_table('pgbench_branches');
select create_reference_table('pgbench_tellers');
select create_reference_table('pgbench_history');
```

```
# on the command line
pgbench -iI gp  -s 300 pgbench
```

```
-- pgcoord1
alter table pgbench_accounts add foreign key (bid) references pgbench_branches;
alter table pgbench_history add foreign key (bid) references pgbench_branches(bid);
alter table pgbench_history add foreign key (bid) references pgbench_tellers(tid);
alter table pgbench_tellers add foreign key (bid) references pgbench_branches(bid);

alter table pgbench_accounts
    validate constraint pgbench_accounts_bid_fkey;

alter table pgbench_history
    validate constraint pgbench_history_bid_fkey,
    validate constraint pgbench_history_bid_fkey1;

alter table pgbench_tellers
    validate constraint pgbench_tellers_bid_fkey;
```

Configure pgbench Tables Cont’d



pgbench_tellers [table]

tid

bid

...

< 1 | 300 rows | 1 >

pgbench_branches [table]

bid

...

30 rows | 3 >

pgbench_history [table]

tid

bid

aid

...

< 3 | 0 rows

pgbench_accounts [table]

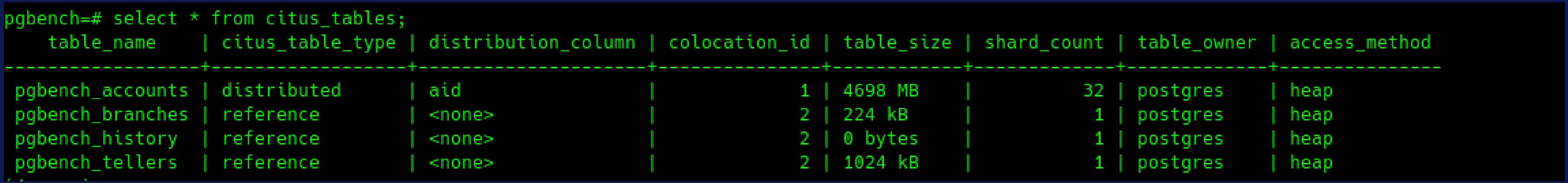
aid

bid

...

< 1 | 3,000,000 rows | 1 >

Configure pgbench Tables Cont’d



```
pgbench=# select * from citus_tables;
    table_name    | citus_table_type | distribution_column | colocation_id | table_size | shard_count | table_owner | access_method
------------------+------------------+---------------------+---------------+------------+-------------+-------------+---------------
 pgbench_accounts | distributed      | aid                 |             1 | 4698 MB    |          32 | postgres    | heap
 pgbench_branches | reference        | <none>              |             2 | 224 kB     |           1 | postgres    | heap
 pgbench_history  | reference        | <none>              |             2 | 0 bytes    |           1 | postgres    | heap
 pgbench_tellers  | reference        | <none>              |             2 | 1024 kB    |           1 | postgres    | heap
```

Summary

```
- get, install
    - postgres
    - citus extension
    - patroni
    - etcd
- configure
    - postgresql.conf
    - pg_hba.conf
    - .pgpass
- create cluster (confirm all nodes can contact each other)
    - coordinator
        - PRIMARY
        - STANDBY
    - worker nodes (3X)
        - PRIMARY
        - STANDBY
    - pgbench: on all nodes
- configure
    - systemd (disable postgres)
    - patroni
        citus:
            group: ${GROUP}
            database: pgbench
- patroni
        - validate
        - test
        - startup
- pgbench
    - configuration
    - data population
```

Questions?

https://github.com/rbernierZulu/pg_conf_Seattle-2024

Thank You!

PASS24
Data Community Summit