**Lab Exercise 18- Failover Management with Patroni and PgBouncer**

**Objective**

* Set up a high-availability PostgreSQL cluster using **Patroni**
* Use **PgBouncer** to handle connection pooling and automatic redirection after failover

**Tools Used**

* PostgreSQL version 14 or later
* Patroni for automatic failover
* Etcd (or Consul) as distributed configuration store
* PgBouncer as connection pooler

**Environment Setup (3 Machines or Virtual Machines)**

| **Component** | **Hostname** | **Role** |
| --- | --- | --- |
| postgres1 | node1.local | Primary Candidate |
| postgres2 | node2.local | Standby Candidate |
| etcd | etcd.local | Key-value store |
| pgbouncer | lb.local | Connection router |

**Step 1: Install Required Packages (Linux Example)**

sudo apt update

sudo apt install postgresql python3 python3-pip etcd

pip3 install patroni[etcd]

Do this on both database nodes.

**Step 2: Configure Patroni (On node1 and node2)**

Create a YAML file patroni.yml with the following:

scope: postgrescluster

name: node1

etcd:

host: etcd.local colon 2379

postgresql:

bin\_dir: /usr/lib/postgresql/14/bin

data\_dir: /var/lib/postgresql/data

parameters:

wal\_level: replica

hot\_standby: on

max\_wal\_senders: 10

max\_replication\_slots: 10

synchronous\_commit: local

authentication:

replication:

username: replicator

password: replpass

superuser:

username: postgres

password: adminpass

restapi:

listen: 0.0.0.0 colon 8008

connect\_address: node1.local colon 8008

tags:

nofailover: false

noloadbalance: false

clonefrom: false

Modify name and connect\_address for node2 accordingly.

**Step 3: Start Patroni on Both Nodes**

patroni patroni.yml

Use curl <http://localhost:8008> to verify the running node.

**Step 4: Install and Configure PgBouncer (On lb.local)**

Install:

sudo apt install pgbouncer

Edit pgbouncer.ini:

[databases]

postgres = host=node1.local port=5432 user=postgres password=adminpass

[pgbouncer]

listen\_addr = 0.0.0.0

listen\_port = 6432

auth\_type = md5

auth\_file = /etc/pgbouncer/userlist.txt

Create userlist.txt:

"postgres" "md5 hash of password"

Start PgBouncer:

sudo systemctl start pgbouncer

**Step 5: Test Failover**

1. Connect to PgBouncer using:

psql -h lb.local -p 6432 -U postgres

1. Shut down node1:

sudo systemctl stop patroni

1. Observe Patroni on node2 automatically promoting itself to primary.
2. PgBouncer will still route to the new master (after TTL).

**Summary**

| **Component** | **Action** |
| --- | --- |
| Patroni | Handles monitoring, failover, promotion |
| PgBouncer | Manages client connections and routing |
| PostgreSQL | Runs the actual database engine |

**Validation Checklist**

* Verify Patroni status using REST API or logs
* Connect to PgBouncer and confirm continuity during failover
* Test automatic failover by stopping the primary