PostgreSQL Linux Command Cheat Sheet (Updated)

1. PostgreSQL Service Management

```
sudo systemctl start postgresql
sudo systemctl stop postgresql
sudo systemctl restart postgresql
sudo systemctl enable postgresql
sudo systemctl status postgresql
```

2. User and Role Management

```
sudo -i -u postgres
psql
createuser <username>
createuser --interactive --pwprompt --superuser <username>
dropuser <username>
```

3. Database Management

```
createdb <dbname>
dropdb <dbname>
psql -c "\l"
psql -d <dbname>
```

4. SQL Execution and Interactive Mode

```
psql
psql -d <dbname> -f /path/to/file.sql
psql -d <dbname> -c "SELECT * FROM table_name;"
```

5. Backup and Restore

```
pg_dump -U <username> <dbname> > backup.sql
pg_dump -U <username> -F c -b -v -f backup.dump <dbname>
psql -U <username> -d <dbname> -f backup.sql
pg_restore -U <username> -d <dbname> -v backup.dump
pg_dumpall -U <username> > all_databases.sql
psql -U <username> -f all_databases.sql
```

6. Database Inspection and Queries

```
\du
\dt
\d <table_name>
SELECT * FROM pg_stat_activity;
SELECT current_database();
```

7. Password and Access Control

```
\password <username>
sudo vi /etc/postgresql/<version>/main/pg_hba.conf
sudo systemctl reload postgresql
```

8. Log and Config File Management

```
sudo vi /etc/postgresql/<version>/main/postgresql.conf
SHOW config_file;
SHOW hba_file;
sudo tail -f /var/log/postgresql/postgresql-<version>-main.log
```

9. Performance and Monitoring

```
SELECT * FROM pg_stat_activity;
SELECT * FROM pg_locks;
SELECT * FROM pg_stat_user_tables;
SELECT * FROM pg_stat_user_indexes;
```

10. Useful Utilities and Maintenance

```
reindexdb -U <username> <dbname>
vacuumdb -U <username> -d <dbname> -v
vacuumdb -U <username> -d <dbname> --analyze
```

11. Performance Tuning Commands

```
SHOW work_mem;
SHOW shared_buffers;
SHOW effective_cache_size;
SHOW maintenance_work_mem;
SHOW random_page_cost;
```

```
SHOW seq_page_cost;
SELECT pid, now() - pg_stat_activity.query_start AS duration, query FROM pg_stat_activity WHERE state != 'idle' AND now() - pg_stat_activity.query_start > interval '5 minutes';
SELECT relname, 100 * idx_scan / (seq_scan + idx_scan) AS idx_usage_pct FROM pg_stat_user_tables WHERE seq_scan + idx_scan > 0 ORDER BY idx_usage_pct ASC;
SELECT schemaname, tablename, reltuples::bigint AS rows, relpages::bigint AS pages, pg_size_pretty(pg_table_size(schemaname || '.' || tablename)) AS size FROM pg_tables JOIN pg_class ON tablename = relname WHERE schemaname NOT IN ('pg_catalog', 'information_schema') ORDER BY pg_table_size(schemaname || '.' || tablename) DESC;
ANALYZE VERBOSE <table_name>;
VACUUM VERBOSE <table_name>;
SHOW autovacuum;
SHOW autovacuum_naptime;
SHOW autovacuum_vacuum_threshold;
SELECT blocked_locks.pid AS blocked_pid, blocked_activity.query AS blocked_query, blocking_locks.pid AS blocking_pid, blocking_activity.query AS blocking_query FROM pg_catalog.pg_locks blocked_locks JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid = blocked_locks.pid JOIN pg_catalog.pg_locks blocking_locks ON blocking_locks.locktype = blocked_locks.locktype AND blocking_locks.database IS NOT DISTINCT FROM blocked_locks.database AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid AND blocking_locks.transactionid IS NOT DISTINCT FROM blocked_locks.transactionid AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid AND blocking_locks.pid != blocked_locks.pid JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid = blocking_locks.pid WHERE NOT blocked_locks.granted;
```